

Semantic Sentence Similarity Incorporating Linguistic Concepts

DAVID MATTHEW PEARCE

A thesis submitted in partial fulfilment of the requirements
of the Manchester Metropolitan University for the degree
of Doctor of Philosophy

School of Computing, Mathematics and Digital
Technology the Manchester Metropolitan University

June 2015

Abstract

A natural language allows a set of simpler ideas to be combined together to communicate much more complex ideas. This ability gives language the potential for use as a highly intuitive method of human interaction. However, this freedom of expression makes interpreting language with automation extremely challenging.

Semantic sentence similarity is an approach which allows the knowledge of how to compare simpler units, such as words, to obtain a measure of similarity between two sentences. This similarity can allow existing knowledge to be applied to new situations.

The objective of this research is to show that a sentence similarity model can be improved through the inclusion of Linguistic concepts, with the aim of producing a more accurate model. This presents the challenge of adapting the human focused rules of Linguistics for sentence similarity and how to evaluate individual component effects in isolation.

This research successfully overcame these barriers through the development of an extensible modular framework and construction of a new mathematical model for this framework, called SARUMAN. The core contribution of the research resulted from gradually incorporating fundamental Linguistic components to SARUMAN including: disambiguation by part of speech; treating the sentence as clauses, and advanced word interaction to handle where meanings merge. The most advanced being called SCAWIT. From experiments on a small data set, each of these introduced concepts showed statistically significant improvement in the Pearson's correlation (0.05 or more) over the previous version. The produced models were capable of processing several hundred sentence pairs a second with a single processor.

A further significant advance to the field of sentence similarity was the introduction of opposites to sentence similarity. This was conceptually beyond the pre-existing models and showed strong results for an extension of SCAWIT, called SANO.

Other novel contribution was added through automated word sense disambiguation from WordNet definitions; and the use of a properties of words model. Some of these changes have potential but did not yield significant improvement with the current knowledge base.

Acknowledgements

I would like to thank anyone has offered me advice or assistance.

I need to give thanks my project supervisors: Dr. Zuhair Bandar and Dr. David McLean for their advice and direction, with special thanks to my Director of Studies, Zuhair, for his dedication and assistance.

I owe a debt to my family for their support, patience and understanding while I undertook this long journey to complete my research and thesis, allowing for an investment in my future.

Statement of Originality

The material in this thesis has not previously been submitted for a degree in any University, and to the best of my knowledge contains no material previously published or written by another person except where explicit acknowledgement and reference is made in the thesis itself.

(David Matthew Pearce)

Table of Contents

1.0 Introduction.....	1
1.1 Introduction.....	1
1.2 Sentences.....	1
1.3 Similarity.....	2
1.4 Objective.....	2
1.5 Motivation.....	3
1.6 What is Sentence Similarity?.....	3
1.6.1 Absolute Semantic Similarity.....	4
1.7 Association versus Similarity.....	4
1.8 Basic Model.....	5
1.9 Novelty.....	5
1.10 Chapter Summaries.....	8
 2.0 Background & Literature Review.....	 11
2.1 Introduction.....	11
2.2 Adapting Linguistics to Sentence Similarity.....	12
2.2.1 Words.....	13
2.2.2 Sentences.....	13
2.3 Challenges when Comparing Sentences.....	14
2.3.1 Topic.....	15
2.3.2 Word Interaction.....	16
2.3.3 Context.....	17
2.4 Grammar.....	20
2.4.1 What is a Clause?.....	21
2.4.2 Functional Purposes of Clauses.....	21
2.4.3 Subject - Verb - Object.....	22
2.4.4 Adjunct Clauses.....	23
2.4.5 Demonstrative and Participle Clauses.....	23
2.4.6 Complex and Compound Sentences.....	25
2.4.7 Parts of Speech.....	26
2.4.8 Direct Objects.....	26

2.4.9 Implied Words.....	27
2.4.10 Perfect Ambiguity.....	28
2.5 Automatic Parsers.....	29
2.6 Knowledge Sources.....	31
2.7 Representations of Meaning and Similarity.....	31
2.7.1 Comparing Structures.....	32
2.7.2 Conceptual Descriptions.....	35
2.7.3 Ontologies.....	35
2.7.4 Sentence Relationships.....	36
2.7.5 WordNet.....	37
2.8 Word Similarity Measures.....	39
2.9 Corpus Based Sentence Similarity.....	41
2.10 Knowledge Based Sentence Similarity.....	44
2.11 Relatedness Measures.....	46
2.12 Limitation of Current Sentence Similarity Models.....	46
2.13 Conclusions.....	48
3.0 Datasets.....	51
3.1 Introduction.....	51
3.2 Words Datasets.....	52
3.3 STASIS-30 Dataset.....	53
3.3.1 Limitations of STASIS-30 Dataset.....	53
3.4 Microsoft Research Paraphrases Dataset (MSRP).....	54
3.4.1 Limitations of MSRP for Sentence Similarity.....	54
3.5 New Datasets.....	54
3.5.1 Design Approach and Constraints.....	55
3.5.2 Method for Generating Pairs.....	57
3.5.3 Human Ratings for Similarity Scores.....	58
3.6 New General Purpose Datasets.....	60
3.6.1 Ten Pairs Dataset Construction.....	60
3.6.2 Ten Pairs Dataset.....	62
3.6.3 Thirty Pairs Dataset Construction.....	64

3.6.4 Thirty Pairs Dataset.....	65
3.6.5 Difference in Duplicated Sentence Scores.....	70
3.7 Opposites.....	74
3.7.1 New Scale for Opposites.....	74
3.7.2 Constructing the Opposites Dataset.....	75
3.7.3 Opposites Dataset.....	76
3.8 Conclusions.....	83
4.0 Linguistic Framework.....	85
4.1 Introduction.....	85
4.2 Framework.....	86
4.3 Word Meanings.....	90
4.4 Context.....	94
4.5 Algorithm Module.....	97
4.6 Module Development.....	99
4.7 Conclusions.....	100
5.0 Experimental Method and Evaluation.....	103
5.1 Introduction.....	103
5.2 Purpose.....	103
5.3 Summary Metrics.....	105
5.4 Experimental Method.....	106
5.4.1 Core Experimentation.....	106
5.4.2 Statistical Significance.....	107
5.4.3 Benchmarking.....	109
5.4.4 Domain Testing.....	109
5.4.5 Opposites.....	109
5.4.6 Conclusions.....	109
6.0 Mathematical Model.....	111
6.1 Introduction.....	111
6.2 Implementation.....	112
6.3 Knowledge Source.....	112

6.3.1 Encoding Meaning Structure from WordNet.....	113
6.4 Word Meaning Similarity Module.....	115
6.5 Weights.....	116
6.6 SARUMAN Algorithm.....	116
6.7 Experiments.....	119
6.8 Benchmarking SARUMAN.....	120
6.8.1 Comparing SARUMAN to other Models.....	122
6.9 Experimental Benchmark.....	129
6.10 Conclusions.....	133
7.0 Disambiguation of Meaning and Type.....	135
7.1 Introduction.....	135
7.2 Disambiguation Approaches.....	135
7.3 Human Tagging.....	136
7.4 Automatic Disambiguation.....	140
7.4.1 Using WordNet's Definitions.....	142
7.4.2 Disambiguation Weighting Module.....	143
7.5 Experiments.....	147
7.6 Results and Discussion.....	148
7.6.1 Human Meanings and POS Tagging.....	153
7.6.2 Automatic Disambiguation.....	154
7.7 Conclusions.....	156
8.0 Disambiguation of Clauses.....	159
8.1 Introduction.....	159
8.2 Parser.....	160
8.2.1 Parser Module Objective.....	161
8.2.2 Differences to General Purpose Parsers.....	162
8.2.3 Simplified Sequential Parser Example.....	165
8.2.4 Limitations of the Parser.....	170
8.2.5 Implementation.....	171
8.2.6 Parser Implementation.....	173
8.2.7 Clause Tagging.....	176

8.2.8 Parser Example.....	177
8.2.9 Parser Results.....	179
8.3 Extending the Vocabulary.....	182
8.4 Cross-type Comparison.....	184
8.5 Clause Disambiguation Implementation.....	185
8.6 Experiments.....	186
8.7 Results.....	186
8.8 Conclusions.....	190
9.0 Combining Properties for Similarity.....	193
9.1 Introduction.....	193
9.2 Handling auxiliary and modal verbs.....	195
9.3 Limitations of Li et al. Formula.....	197
9.4 Properties of Words Model (PoW).....	199
9.5 Example of a Clause Comparison.....	200
9.6 Experiments.....	202
9.7 Comparing Word Models with Merging.....	204
9.8 Results.....	205
9.9 Conclusions.....	211
10.0 Advanced Word Interaction.....	213
10.1 Introduction.....	213
10.2 Triangulation.....	215
10.3 Alignment.....	216
10.4 Order of Calculation.....	217
10.5 Weights.....	218
10.6 SCAWIT Algorithm Module.....	220
10.6.1 Limitation.....	225
10.7 Experiments.....	225
10.8 Results.....	226
10.9 Conclusions.....	230
11.0 Discussion and Timings.....	234

11.1 Introduction.....	234
11.2 Discussion of core experiment.....	235
11.3 Human scores.....	237
11.3.1 Pair 3.....	238
11.4 Timings.....	241
11.4 Conclusions.....	242
12.0 Benchmark: Thirty Pairs Dataset.....	244
12.1 Introduction.....	244
12.2 Results.....	245
12.3 Conclusions.....	252
13.0 Paraphrase Domain.....	253
13.1 Introduction.....	253
13.2 Method.....	254
13.3 Mean Similarity and Threshold.....	254
13.4 Paraphrase Identification.....	255
13.4.1 Specialist Methods.....	258
13.4.2 SCAWIT.....	259
13.5 Errors in the MSRP dataset.....	259
13.6 Issues with SCAWIT.....	262
13.7 Saturation of Similarity.....	263
13.8 Future Work.....	264
13.9 Conclusions.....	266
14.0 Handling Opposites.....	267
14.1 Introduction.....	267
14.2 New Similarity Scale.....	268
14.3 Types of Opposites.....	268
14.4 Word Meanings for Opposites.....	269
14.5 Word Meaning Similarity Module.....	270
14.6 Opposite Verb Clauses.....	271
14.7 Other Opposites.....	272

14.7.1 Inversion.....	273
14.8 SANO Algorithm Module.....	275
14.8.1 Identifying Negation and Inversion.....	277
14.9 Results.....	278
14.10 Conclusions.....	283
15.0 Conclusions.....	287
15.1 Introduction.....	287
15.2 Contributions.....	291
15.3 Key Findings.....	294
15.4 Future Work.....	296
15.4.1 Tagged Corpora.....	297
15.4.2 Database of Properties of Words.....	298
15.4.3 Greater Collaborative Resources.....	298
References.....	299
Glossary.....	310
Appendix: Related Refereed Publications.....	312

List of Tables

Table 3.1: Ten pairs dataset and the mean of the human score.....	63
Table 3.2: The individual human ratings the thirty pairs dataset.....	64
Table 3.3: Thirty pairs dataset and the mean of the human scores.....	66
Table 3.4: The individual human ratings the thirty pairs dataset.....	69
Table 3.5: The opposites datasets with individual human scores and mean	77
Table 6.1: Leading sentence similarity models' correlations for STASIS-30.....	123
Table 6.2: Numerical values of SARUMAN, STASIS and LSA for STASIS-30	124
Table 6.3: SARUMAN, LSA, OMIOTIS & STASIS for ten pairs dataset	129
Table 6.4: Ten Pairs Dataset and SARUMAN scores	132
Table 7.1: Human tagging for part of speech and meaning for ten pairs dataset	138
Table 7.2: Numerical values for SARUMAN with disambiguation of meaning	148
Table 7.3: The correlations for SARUMAN with disambiguation of meaning	149
Table 8.1: The possible sequences forming clauses for the example in fig. 8.1	167
Table 8.2: The comparison for valid patterns for an article followed by J/N.....	167
Table 8.3: The test for possible patterns from table 8.2 when adding a verb.....	168
Table 8.4: The outcome of adding the parts of speech to an article.....	168
Table 8.5: The outcome of adding the parts of speech to an article + noun	169

Table 8.6: Adding a verb to: article + J/N from sequences in tables 8.4 & 8.5	169
Table 8.7: The single state output for adding possible part of speech when treating article adjective/noun as a single sequence	170
Table 8.8: Possible part of speech ids used by the sequential parser	172
Table 8.9: All possible internal states for the sequential parser	175
Table 8.10: Clause type ids and descriptions for the sequential parser	176
Table 8.11: Demonstration of flow of internal states (from table 8.9) and actions	178
Table 8.12: Results of sequential parser for ten pairs dataset	179
Table 8.13: Summaries for SARUMAN + cross-type comparison	186
Table 9.1: O-P-T-I-C properties for auxiliary verbs	196
Table 9.2: Effect of primary modal verbs and tense to O-P-T-I-C	196
Table 9.3: Classification accuracy for SARUMAN for different word similarity modules on the Mitchell and Lapata (2010) dataset	205
Table 9.4: Summary values for SARUMAN with PoW the ten pairs dataset	206
Table 9.5: Numerical results for SARUMAN with PoW on the ten pairs dataset	207
Table 10.1: Numerical values for SCAWIT using the ten pairs dataset	226
Table 10.2: Summary information for the ten pairs dataset	227
Table 11.1: Summary information for ten pair dataset with an adjusted pair 3.....	240
Table 11.2: Timings for SARUMAN and SCAWIT	242
Table 12.1: LSA and SCAWIT for thirty pairs dataset.....	245

Table 12.2: Summaries for LSA and SCAWIT on the Thirty Pairs dataset.....	249
Table 13.1: Similarity models paraphrase identification scores for MSRP test set.....	256
Table 13.2: Supervised specialist paraphrase models and others with MSRP.....	257
Table 14.1: SANO results on the Opposites Dataset	279
Table 14.2: Summaries for SANO and opposite tagged LSA and SCAWIT.....	283

List of Figures

Figure 2.1: Possible meaning structures pairs.....	33
Figure 4.1: Linguistic Sentence Similarity Framework	88
Figure 4.2: Knowledge Base Modules	90
Figure 4.3: An illustration of the weights for a pair of sentences	93
Figure 4.4: Context and Weighting Modules	95
Figure 4.5: Showing how the weights would be combined for two words	96
Figure 6.1: WordNet hypernym chains for “Cat” and “Alsatian”	113
Figure 6.2: SARUMAN on the STASIS-30 dataset	121
Figure 6.3: SARUMAN (without weights) vs. SARUMAN for the STASIS-30	122
Figure 6.4: SARUMAN vs. STASIS for the STASIS-30 dataset	125
Figure 6.5: SARUMAN vs. LSA for the STASIS-30 dataset	126
Figure 6.6: SARUMAN vs. IISIS for the STASIS-30 dataset	126
Figure 6.7: SARUMAN vs. OMIOTIS for the STASIS-30 dataset	127
Figure 6.8: SARUMAN versus DTW for the STASIS-30 dataset	127
Figure 6.9: SARUMAN vs. SyMSS for the STASIS-30 dataset	128
Figure 6.10: LSA with Ten Pairs dataset	131
Figure 7.1: Possible meanings of example three words in a sentence	145

Figure 7.2: Weights for comparing words A& B from figure 7.1	145
Figure 7.3: Adding Weights for comparing words A& C to figure 7.2	146
Figure 7.4: Final disambiguation weights for diagram 7.1	147
Figure 7.5: SARUMAN with no disambiguation	149
Figure 7.6: SARUMAN with disambiguation of type	150
Figure 7.7: SARUMAN with human tagged meaning disambiguation	151
Figure 7.8: SARUMAN with automatic disambiguation of meaning	152
Figure 7.9: SARUMAN with automatic disambiguation filtered by tagged type	152
Figure 8.1: The four valid patterns for the simple example for sequential parser	166
Figure 8.2: SARUMAN with cross-type comparison	188
Figure 8.3: SARUMAN with type disambiguation and cross-type comparison	189
Figure 8.4: SARUMAN with human meanings and cross-type comparison	189
Figure 8.5: SARUMAN with clause disambiguation	190
Figure 9.1: Contribution of common meaning to the Li et al. formula	197
Figure 9.2: Contribution of Distance between meanings to Li et al. formula	198
Figure 9.3: SARUMAN with PoW	208
Figure 9.4: SARUMAN with PoW and type disambiguation	209
Figure 9.5: SARUMAN with PoW, OPTIC and clause disambiguation	210
Figure 9.6: SARUMAN with clause disambiguation and properties by hand.....	210

Figure 10.1: Importance weights for two PSVO clauses	219
Figure 10.2: The importance weights for two PSVO clauses aligned	219
Figure 10.3: The importance weights distributed to individual clauses for two PSVO clauses aligned.....	220
Figure 10.4: SCAWIT using Li et al. (2003)	228
Figure 10.5: SCAWIT with PoW	229
Figure 10.6 Properties by hand for the ten pairs dataset with SCAWIT	230
Figure 12.1 LSA for thirty pairs dataset	250
Figure 12.2 SCAWIT on the thirty pairs dataset	251
Figure 14.1: SANO for the opposites data set.....	282

1.0 Introduction

1.1 Introduction

This chapter outlines what precisely defines semantic sentence similarity and why it is an important area of research. The objectives and motivation of the research are set out alongside the key areas of novelty and brief summaries of the subsequent chapters in the thesis.

It has been a long-sought goal to produce a machine that could interact with a person using their everyday language (Maynard et al., 2004) (Zhongzhi, 2006). A natural language makes complex communication possible because it provides a common set of terms (words) and rules (grammar) which can be combined to express more complex ideas.

1.2 Sentences

The sentence is a fundamental unit of language (Huddleston et al., 2002) is able to carry a meaning beyond that of the words which the sentence comprises. A sentence allows not only for more refined meanings than words to be conveyed but also meanings beyond those which words alone could convey. (Huddleston et al., 2002) (Quirk, 1962)

A natural language, such as English, offers the potential for a very powerful interface with the freedom for people to intuitively communicate their wishes with many alternative ways to express any idea. Unfortunately, it is this freedom of expression that makes it difficult to have a machine interpret the intended meaning from a natural language input; due to the near infinite number of possible sentences and meanings.

1.3 Similarity

What is meant when something is described as being similar to something else? This is a question that is crucial when making any comparison. It allows known information to be applied to new situations (Goldstone and Son, 2005) and is one of the fundamentals of intelligence (Markman and Gentner, 1993).

When stating that two things are alike, it means that there are common attributes between them. If it is known which of the attributes are the same then the information only needs to be known about one, to apply it to both. This allows for characteristics to be generalised. Generalisation not only allows information to be stored more efficiently but for new things to be compared against existing knowledge. When things are compared with other things people look for similarity between the items (Larkey and Markman, 2005) . It is these characteristics of similarity that are exploited for sentence similarity.

1.4 Objective

There are many observations in Linguistics that have been made in order to describe English and how people use, construct and interpret language.

The objective of this thesis is to adapt concepts from Linguistics to give a set of rules which can be incorporated into a sentence similarity model, to produce a more accurate sentence similarity model.

The aim is to both evaluate the efficacy of these methods and hopefully to produce a superior sentence similarity model which can realise some of the Linguistic potential for real-time computer applications. Meeting real-time means the ability to process a sentence in the same time as it would to type another one. Since sentence similarity can be processed in parallel for each sentence pair, this requirement would enable multiple comparisons to be done in real time by using more processors.

1.5 Motivation

It has already been stated why sentence similarity is an important tool for the potential automation of the handling of English. While there have been several models (detailed in section 2.9 & 2.10) previously constructed that are capable of judging the semantic similarity of a pair of sentences, none of these have pursued a Linguistic approach. Yet, it is the field of Linguistics that has specialised in studying language and how the parts of language (such as words and clauses combined together) form the more complex ideas needed for communication.

There are many observations from Linguistics about English which affect how the meanings of sentences would compare. Whilst rules from these observations have been developed, the focus has been on human understanding of the rules. However, if these rules could be automated and introduced to sentence similarity it could enable a significant improvement in accuracy beyond what has been achieved to date.

It is the hope of this research that the fundamental rules from Linguistics could be incorporated into a sentence similarity model and this be shown to give an improvement in accuracy as set out in section 1.4.

1.6 What is Sentence Similarity?

Semantic sentence similarity can be simply defined as:

The measure of similarity between the meanings of a pair of sentences.

This thesis is focused on semantic sentence similarity for written English. While a sentence has no formal upper limit of length (McArthur (ed.) , 1991) the interest for this research is predominantly for sentences with one main verb which would correspond to lengths of

input sentences probably in the range of 3 to 20 words. Below this length would be of interest to word similarity and above this length could be heading towards document length. However, the sentence similarity model can cope with inputs n at the word length and with longer inputs provided that it is a single sentence per input.

Malformed sentences or even completely random sequences of characters are both valid inputs and the model should return a value for these inputs. However, the greater the number of errors in the input data, the less meaningful any similarity score will likely become as a sentence similarity model assumes correctly formed language for the input.

1.6.1 Absolute Semantic Similarity

Every input sentence can be thought of as representing a specific meaning. The meanings of two sentences will have a fixed semantic similarity. This score might not be known or easily obtainable but it is reasonable to assume that every pair of sentences has an ideal fixed similarity score that could be regarded as its actual similarity. This, ideal similarity for a pair of sentences, is its absolute semantic similarity.

1.7 Association versus Similarity

There are other ways that sentences can be connected to one another beyond simply considering their meaning. Examples include identifying questions and answers (De Boni and Manandbar, 2003); and relatedness which looks for how sentences logically connect. These other methods of comparing sentences can cause confusion with the meaning of similarity. The focus of this research is only on the similarity of meaning and other relationships or associations are only of interest as to how they are being applied to semantic sentence similarity.

1.8 Basic Model

In its simplest form, a sentence similarity model takes two inputs of text (which ideally would be sentences) but could be any possible input. The premise is that there is an absolute semantic similarity which can be attributed to any pair of sentences which would be the idealised output of the model.

Because sentence similarity works by comparing general concepts, there is no exact calculation available to find the absolute similarity. Essentially the sentence similarity model is only an approximate method.

1.9 Novelty

Existing models can be broadly split into two categories of corpus methods (section 2.9) and knowledge based methods (2.10) where a lexical database is used.

They have built a set of concepts for each word based upon the co-occurrence of words in large corpora. These concepts can then be used to produce a vector of concepts for each vector which can be combined to give a single value for the overall similarity of the two vectors. The main corpus methods relevant to sentence similarity are: Latent Semantic Analysis which has been one of the most significant sentence similarity methods. More recent efforts have introduced a word order component - IISIS (Islam and Inkpen, 2007); or in the use of structured knowledge for its corpus such as Wikipedia with ESA and SSA,

The main knowledge based methods of sentence similarity use the Lexical database WordNet's ontological relationships (Feldbaum (ed.), 1998) (section 2.7.5) in order to judge the similarity of a pair of sentences. Each word in the sentences is compared individually with the words in the other sentence to find the highest similarity. A combination of the individual scores for each word and the corresponding position of a word in the other sentence which the highest similarity was obtained have been used to

obtain a single value for the similarity of the sentence pairs. The main knowledge based methods are STASIS (Li et al, 2006) which uses just the nouns from WordNet and word order similarity; Sentence similarity with Dynamic Time Warping (Liu et al., 2007) and OMIOTIS (Tsatoronis et al., 2010) which uses all of the ontological relationships in WordNet. More recent changes to knowledge based sentence similarity have included adding Fuzzy word handling to STASIS (Chandran et al., 2013) and SyMSS which included a parse tree (Oliva et al. 2011).

The objective of this thesis is to adapt concepts from Linguistics and incorporate them into a sentence similarity model. While some of the pre-existing sentence similarity models contain elements which relate to Linguistic concepts, in essence they can be thought of as mathematical models where every word is treated the same within a sentence (irrespective of its Linguistic function.)

There were several challenges to meeting the objective. These included the fact that Linguistics has developed its rules so that a person can understand the meaning of a single sentence and these need to be adapted for sentence similarity. It is also not the case that you could create a sentence similarity model that could be regarded as complete from a Linguistic approach because fundamentally Linguistics is based upon observations of Language and not a finite set of rules.

This means that there is the need to experimentally isolate the contribution from individual Linguistic concepts to a sentence similarity model. While the existing mathematical models can show a solid performance for the task of sentence similarity, it is not possible to simply add Linguistic ideas to them as their architecture does not allow for extension. In addition, many of the models contain components that are inherently incompatible with a Linguistic approach that prohibit r-using many attributes even with an improved architecture (section 2.11).

The final challenging issue is that identifying a Linguistic attribute as important is not the same as being able to implement it so that an improvement in similarity will automatically result. Many of the tasks needed for sentence similarity can only be approximate and the aim to produce a more accurate sentence similarity model still has to outperform the solid performance from some of the mathematical models.

These were shown to be capable of processing 100s of pairs of sentences a second (Section 11.3) with the potential to be more accurate than most of the pre-existing methods because of the Linguistic approach. SANO represented the only sentence similarity model currently able to handle opposites as part of its similarity score.

It was proposed to overcome these challenges through adopting a Linguistic approach, that started with identifying the core Linguistic components that apply to sentence similarity (section 2.3) and then using these to build a novel Linguistic framework (chapter 4) that would allow for the gradual introduction of these core concepts to a sentence similarity model.

The research presented in this thesis makes significant contribution to the field of sentence similarity. The Linguistic approach itself had never been adopted for sentence similarity and in meeting the objective laid out in section 1.4. In addition to the experiments, 3 new sentence similarity models were produced called SARUMAN, SCAWIT and SANO.

The core aspects of Novelty of this research are:

- First to adopt a linguistic approach to sentence similarity
- Creation of an extensible sentence similarity framework based upon and capable of gradually adding extra Linguistic components.
- A new sentence similarity model that includes Linguistic ideas that is capable of outperforming many of the existing sentence similarity models.
- First to investigate disambiguation of meaning to an ontological based sentence similarity model.
- Introduction of both disambiguation by part of speech and clauses to sentence similarity model.
- Created a new sentence similarity scale for opposites and adapted a sentence similarity model to handle opposites on this scale.

Other areas of novelty related to the creation of the sentence similarity models although with less significance to the field of the core research:

- Creation of 2 small new datasets with potential use for extra benchmarks.
- A new mathematical sentence similarity model to combine the underlying word similarities to a single measure (Chapter 9).
- The creation of a sequential deterministic parser that produces a specialist output for part of speech and clause tags, (Chapter 8) capable of performing substantially faster than real time performance.
- A proposed new word meaning representation using properties of words and formula for comparing words using properties.

1.10 Chapter Summaries

Chapter 2 gives an overview of the existing work in sentence similarity and the related areas. Additionally, it covers how some key Linguistic concepts can be adapted for the task of comparing a pair of sentences.

Chapter 3 describes the datasets that are to be used as part of the experimentation and the methodology use to create some small new datasets.

Chapter 4 gives the details of the modular framework which is the foundation of having a sentence similarity model which can be evolved to include increasingly sophisticated Linguistic features.

chapter 5 details the experimental methodologies.

Chapters 6-11 form the key part of the experiments. First benchmarking a mathematical model against the existing leading models and then gradually enhancing the model with an additional Linguistic concept and comparing it against the performance prior to the inclusion.

Chapter 12 details a final benchmarking experiment upon the final evolution of the sentence similarity model, from the core experiments, on an enlarged version of the dataset to demonstrate repeatability.

The final two chapters, *13 and 14*, are testing the sentence similarity model in specialist domains of paraphrases and opposites. Where opposites is a previously unresearched area for sentence similarity.

2.0 Background & Literature Review

2.1 Introduction

This chapter gives the background of the current state of the sentence similarity and also outlines key areas that closely relate to the creation of the sentence similarity model. Also included is the adaptation of the core Linguistic concepts that are key to the construction of the framework, for use with the similarity models given later in this thesis. Additionally, these concepts highlight how a sentence similarity model could be enhanced through the inclusion of these concepts.

Understanding human language has been an important part of the goals of AI since its inception. One of the classic tests as to whether a machine has achieved intelligence is the Turing test (Turing, 1992) where a machine has to hold a conversation which cannot be distinguished from a person. There has followed a steady set of research into spoken language from SOUNDEX (Kempken et al., 2006) to modern applications such as SIRI (Bellegarda, 2014).

There are several areas which relate to the research to be presented in this thesis from both Linguistics and computing. Linguistics is far too broad a topic to be fully covered here. First the core concepts of Linguistics are defined and adapted for the purpose of sentence similarity. While the ideas in Linguistics were well known, Linguistics is interested in the meanings of individual sentences not comparing the similarity of the meanings of pairs of sentences. Therefore, identifying how these rules can be used for comparing sentence is a novel step and they form the basis of the framework used for building the sentence similarity models used in this research.

There are a large number of rules of grammar and methods to classify the function of each component and word in a sentence as they combine to form the overall meaning of the sentence. The key components that are needed to be understood to parse a sentence are given in section 2.4 with examples from English to illustrate the ideas in practice.

An overview of parsers is given in the next subsection as the task of parsing becomes pertinent as the more complex areas of word interaction are introduced to a sentence similarity model in from chapter 7 onwards.

The rest of this chapter discusses the tools that are used to compare language with increasing complexity. A brief overview of the main sources of knowledge that are used and representations of meaning and similarity relationships are given before discussing how pairs of words can be compared.

This cumulates in an overview of the leading models that have been used for sentence similarity in section 2.10 and specialist models in 2.11, followed by a section analysing their limitations with respect to the Linguistic ideas set out earlier in the chapter.

2.2 Adapting Linguistics to Sentence Similarity

There are many observations in Linguistics that have been applied to finding and understanding the meanings of sentence. However, for sentence similarity and automation many of these observations need to be adapted for the context of comparing the meanings of pairs of sentences.

2.2.1 Words

To compare sentences requires first the ability to compare words. Most of the terms that are needed to discuss this already exist in English and are well understood but their ubiquity results in ambiguity when using words such as 'word' or 'sentence'. Other terms are introduced in this section due to their importance to comparing the meaning of sentences. When discussing words the following nomenclature is going to be adopted:

- *Form* - a word's form is how the word is spelt.
- *Meaning* - What idea the word is referring to.
- *Stem* - the base word which a suffix has been integrated to give the form.
- *Suffix* - a morpheme that has been appended to the end of the stem.
- *Type* - the "part of speech" that a stem has.

A stem can have several different suffixes added, each yielding a different form. A form of the word can have more than one possible meaning and even more than one stem and type. In contrast each meaning has a unique stem and a single type.

So consider the word 'cats', the form is simply how it is written 'c', 'a', 't', 's' or 'cats'.

The stem is 'cat' and hence the suffix is just the letter 's'. The stem exists with more than one type and in the case of cat this can be either a verb or a noun. It can have several possible meanings such as "more than one feline" or alternatively "whip". It could be the third person singular of verbs meaning "to whip" or "to vomit".

2.2.2 Sentences

A sentence is one of the fundamental units of grammar and in written form it is clearly indicated through the use of punctuation. As with words, in order to discuss the similarity of sentences some terms need to be defined.

Every sentence has a topic and a meaning:

- *Topic* - The topic of a sentence is the subject which it is discussing. This is the information that is stored in an index. In an index, a keyword tells you where to find the information that pertains to the keyword. It does not, however, tell you what that information is going to say.
- *Meaning* - This describes the idea that the sentence is conveying as whole. This is not just what the sentence is about, but the idea and action to which the sentence refers.
- *Word Interaction* - How the words combine to give the meaning of the sentence beyond the meanings of its words and hence beyond the topic.
- *Context* - How the meanings of the words are fixed from the surrounding words. This can be very complex as it also depends upon the ideas being discussed.

Some of these terms may not yet be clear but examples in the next section should make the distinction clearer.

2.3 Challenges when Comparing Sentences

Sentence similarity aims to find the similarity without the need for the complexity of finding the individual meanings of the sentences. This section will examine how the sentence properties introduced above can be used to achieve this aim and show if a sentence similarity model incorporates topic, word interaction and context, then the need to find and directly compare sentences is removed. Additionally, a more detailed explanation of the Linguistic concepts and why they are needed for accurate sentence similarity will follow in this section.

2.3.1 Topic

A clear distinction between topic and word interaction is needed before they can be used for sentence similarity. The topic of a sentence is what a sentence is about. Sentences can have different meanings and share a common topic. The topic of the sentence is perhaps best illustrated when considering indexed data. Take a simple query wishing to know the age of the earth. If the following question were to be placed into a search engine:

"How old is the Earth?"

A search engine would pick out the keywords "old" and "earth". All of the answers should share the topic 'the age of the Earth' and all of the results would contain at least one of the keywords (Baeza-Yates and Ribeiro-Neto, 1999) found in the sentence but need not have a common meaning. Indeed, taking two fragments from this query to such a search in Google (Google website, 2011)

"indicates the earth is less than 10000 years old"

"tell us the earth is 4.7 billions years old"

Both fragments are an answer to "What is the age of the Earth?", but have different meanings. It is possible to further distinguish these results of the search by just looking for differing keywords in the answers. Significant difference can be identified between "10000" and "4.7 billions". This difference in keywords illustrates that the answers have different topics to each other. Despite both being related to the same question as defined in section 2.3.2, the difference in keywords means that they could have differing meaning. Essentially, the topic shows a difference in the meaning of the component words before relying on how the words interact to form more complex meanings. It shows that topic is dominated by the keywords and could be described in terms of the words alone, ignoring any word interaction that make up the meaning.

This means that the topic of sentences can be compared by considering their words individually and then combining this information in to a single measure. So the similarity

between the topics of sentences can be broken down into combining the similarity of words.

2.3.2 Word Interaction

When all of the words in a sentence are identical to the words found in another sentence then their topic must be the same. Consider the following sentence pair which show that even when there is an identical topic that there can be a distinct difference in meaning. Two sentences both describing an outcome of a football match:

"The blues beat the reds to win the final."

"The reds beat the blues to win the final."

With identical words neither keyword disambiguation nor topic evaluation are able to distinguish these two sentences. The topic for both is about if the reds or the blues won the final. Yet, the sentences have a clear difference in meaning.

The meaning of a sentence depends on more than just the meanings of the words which they comprise but also on how the words interact with one another (Quirk, 1962). This extra property of word interaction does not depend exclusively upon word order, as shown by this next sentence pair:

"The dog was eating."

"The dog was eaten."

Here a single pair of words, with the same stem word, change the operation of the rest of the sentence. This operation is determined through the rules of grammar and varies between languages.

The influence of the phrase "the dog" has been altered so as the words interact completely differently with the verb "eat". Word interaction can be described by set rules from Linguistics where the words can be split into clauses. In the first example the subject and

object clauses have been reversed by the words' placement to the main verb. It is the relationship of the other words with the verb that is often most significant to word interaction as can be seen in the above sentence pair.

The change between the function of "the dog" as subject and object is the result of the verb being in a passive mood expressed by the change in the from the present participle ("eating") to the past participle ("eaten"). (McArthur (ed.), 1991). "Eating" because it was an active verb could have appended "the food" as an object clause where as "eaten" already had an object so "the food" could not have been appended and still had a sentence.

2.3.3 Context

Although topic and word interaction together form the meaning, context is also needed to achieve sentence comparison because the meanings of words are affected by their usage. Words can have more than meaning for a given form. With multiple possible meanings of its words, a sentence has a range of meanings rather than a unique meaning. The ambiguity can be resolved if a specific meaning is selected for each of the words. A person selects a single meaning for each of the words based on the context to give a single meaning for the sentence. The context is set by how the ideas that the meanings represent interact with each other.

The meaning of a word becomes fixed from its context and so is influenced by the words that surround it. The context can be influenced by the word interaction but more often it is the underlying ideas that have greatest effect. For example, the word "colon" has three meanings that will often be set by the context of what is being currently discussed. In Linguistics it is much more likely to mean the grammatical mark; in medicine the organ; and in a shop in Ecuador, the currency.

Context is more complex than either word interaction or topic, it is how ideas themselves interact rather than the words that are used to abstract them. In the following analysis, it is not just the complexity of context but also the level of complexity of language as a whole that has to be considered.

A person has the ability to consider the possible meanings of a sentence and select the most appropriate meaning even without knowing the surrounding sentences. Thus sometimes the meaning of a sentence which a person has chosen is altered by later information. A common example of this is misdirection used in humour:

"A man walked into a bar. Ouch!"

The first sentence or clause would normally be setting the situation of a man entering a pub and this is the meaning that the listener is supposed to assume. The humour comes when the second part is encountered with the word "ouch". All of a sudden, the meaning of the phrase "walked into" has changed from 'entered into' to 'bumped into' and the word "bar" would now refer to a 'solid pole'. This example shows that a sentence still has a range of meanings to a person but that they chose a specific meaning. The context is what a person uses to pick the meaning that they believe to be intended or most likely. It also shows how refining the context can change the choice of meaning for the sentence.

The problem of selecting meanings of words from context becomes more difficult in the following example:

"Jean, the carpenter, always makes the bed in his hotel room"

In English, "Jean" would normally mean that the named person was female but in conjunction with the word 'his', the meaning probably has changed to a masculine name from the French. It is still possible that the "Jean" refers to a woman and that the "his" refers to the subject of an earlier sentence but without further context the "his" will be assumed to belong with "Jean".

Here the phrase "makes the bed" is idiomatic but the word "carpenter" would increase the likelihood that the phrase "makes the bed" means "constructed the bed" rather than the idiom from the association of ideas. However, the words "always" and "hotel room" fix the meaning to the idiom. It is not just the ideas the words convey but it is the association made from these ideas. Constructing a bed requires time and resources and a carpenter might have constructed the bed in a particular hotel room, but the idea that a carpenter

would be allowed to and choose to construct a bed every time he slept in a hotel room is so unlikely as to be dismissed.

It is possible that the meaning might be further fixed with more information in the other surrounding sentences but taken in isolation, a human reader would only pick a single meaning. Similarly, the idea of a bed being constructed is less likely in a hotel room than in a workshop. If it had instead said:

"Jean, the carpenter, had made the bed in his hotel room."

Changing "make" to the past tense and removing the word "always" has a dramatic effect on the probable and therefore interpreted meaning. The time taken to construct a bed is less significant if it happened only once and the duration of construction could have been longer. The inclusion of the clause, "the carpenter", if it is not to distinguish this Jean from another Jean would be redundant, unless it were to remove the idiomatic meaning of "making the bed".

Very minor changes in the structure of the sentence can be dramatic to the meaning of the words and hence the sentence as a whole. Although grammatically the words are interacting in the same manner, the change in the ideas is dramatic. This illustrates that to interpret the meaning of a sentence can require not only an understanding of what idea is represented by each word but how the ideas interact with each other in the general experiences of mankind. A task that cannot be accomplished with a simple set of rules.

As language can be used to describe highly complex human ideas (Zwaan, 1999), it is not possible for a model to understand all of the ideas at the sentence level without understanding all of the ideas that those doing the communicating are aware of, not just the rules of grammar. There can also be weaker word interaction between the words from their context such as in the case with:

"A house is on fire."

"The house is on fire."

Just changing the article affects the context of the meaning of the word "house". The

generalisation caused through the use of the indefinite article reduces the impact of the statement. "A house" is much more detached from the listener than "The house". This implies that it is a house that they are familiar with either from the earlier conversation or with no further information they would assume that it is the house that they are currently in. So even a word that has little semantic meaning such as "a" on its own can have a significant impact on the relative importance of the other words.

Context can be seen to be so complex that the information can never be fully fixed and can even be ambiguous for people. The scope of the meanings can be fixed and adding extra information can improve the resolution of the possible meaning to a specific meaning. As the task for which sentence similarity is being used will have a context of ideas, it is desirable to have the context separate from the meaning similarity. The context is needed to estimate the intended meaning of each word but any word can affect the significance to the overall meaning of each word.

2.4 Grammar

Any English speaker should be able to identify the ideas from the examples about to be given in this section. Even without consciously knowing the terms, the ideas referred to can be readily used. There are many books which provide a comprehensible detailed analysis of the basic parts of speech and punctuation such as Taggart and Wines (2008). A few more complex ideas are needed as part of considering the task of parsing a sentence and an overview of these can be found in the Oxford Companion to the English Language (McArthur (ed.), 1991).

The grammar of the sentence is important to identify the word interaction as it defines how the meanings combine together in order to form more complex ideas than could be made with the fixed meaning alone (Quirk, 1962).

2.4.1 What is a Clause?

A clause refers to a group of words that have a specific function in a sentence with respect to forming the meaning of a sentence and can vary in length from a single word to an entire sentence. As a result complex clauses can be subdivided into simpler clauses.

2.4.2 Functional Purposes of Clauses

Clauses can be thought of as fitting into one of three general purposes: descriptive, transformational and dependent.

A descriptive clause adds detail to an object that the listener is familiar with but it does not alter the object. Such as the prepositional clause “on the corner” in the sentence "the car on the corner is mine."

A much more powerful clause is the verb clause which allows for consequences of actions to be described and can be transformational.

"The man killed the thief."

Now, this is no longer a simple description of ideas because there are potential consequences of the action. Firstly, it would be possible to assign states to "the man" as being a killer and "the thief" as being dead. Not only have these states changed, there are also additional consequences as in that it would not be possible for the man to perform the identical action again.

Dependent clauses would be transformational dependent on future factors. "Make me a cup of tea please," because the action of a cup of tea being made only occurs should the listener decide to perform the action upon instruction, would be a dependent clause.

2.4.3 Subject - Verb - Object

The main clause is used to distinguish the core clause from clauses which are subordinate to it (McArthur (ed.), 1991). It is convenient to think of the main clause as comprising a subject clause, verb clause and an object clause. A main clause does not have to contain these three clauses and can be extended with other clauses or a subordinate clause.

The verb clause is essential to allow for the the function of a transformational clause and describes the action taking place and the temporal shift relative to the speaker of the action (past, present or future).

As an example a simple sentence is divided using {} to indicate the clauses:

{*The man*} {*killed*} {*the thief*}.

"killed" is the *verb clause* describing the action to "kill" and signifying that it is an event that has already completed through the use of the past tense.

"The man" is the *subject clause* because he is performing the action.

"the thief" is the *object clause* because the action was performed on the thief.

The subject and object are distinguished from their relative positions to the verb clause and while the subject-verb-object order (SVO) is the normal format in English and so could be described as an SVO language, it is perfectly valid to see the order of the clauses altered. (Taggart and Wines, 2008).

Where the SVO order is changed, this is either from the verb clause using a passive voice so that the subject of the verb functions as the object or from changing the order of the clauses for emphasis. This change of order requires a pause in speech to let the listener know that the order has been changed and in written grammar this pause will normally be indicated with punctuation.

2.4.4 Adjunct Clauses

The basic clause of subject - verb - object can be extended through the addition of other clauses that function as descriptive clauses. The simplest subject and object clauses are noun clauses because their meaning could have been represented by an idea equivalent to ideas that could be represented by a single noun.

Just as a noun can have its meaning enhanced by the addition of an adjective so too can a noun clause have its meaning enhanced by an adjectival clause. An adjectival clause acts on a single noun clause and normally immediately follows the noun clause on which it is operating but can be placed within a noun clause when comma separated.

"He found a ruby the size of his fist."

In the above sentence, "the size of his fist", is an adjectival clause, which functions on the noun "ruby" as if it were an adjective.

Just as the adjectival clause acts on a noun clause, there too is an adverbial clause which acts on the verb clause. The adverbial clause has much freer location in the sentence relative to the main clause but otherwise has very similar structure to a adjectival clause.

"It leapt the fence like a horse."

In this instance "like a horse" functions like an adverb and could have been a single word like "cleanly".

2.4.5 Demonstrative and Participle Clauses

Description can also want to include a verb clause. Whereas normally a verb clause is transformational possibly affecting a change, when the verb clause is part of a participle clause, it merely reports the description of an action whose consequences are already

known or knowable.

"The lady holding a rose is a spy."

Here, the core SVO clause is: "The lady is a spy" the other clause "holding a rose" is a participle clause. In this instance, it functions on the subject effectively as "the lady holding a rose," however, in order to operate as a sentence it would require the addition of another word "is":

"The lady is holding a rose."

Interestingly, this sentence could be formed to include similar information to the first sentence:

"The lady spying is holding a rose."

The difference is in the latter example that it was already known that the lady was spying / a spy but it was not known that she was holding a rose. Conversely, in the former example, it was known to the listener that there was a lady holding a rose but not that she was a spy.

The main clause is adding extra information to the listener where a participle clause is referring to known information. Indeed, participle clauses can be viewed as a type of finite subordinate clauses (McArthur (ed.), 1991).

The subordinate clause, like the main clause, adheres to the SVO model. With the participle clause, its subject is shared with a noun clause in the main sentence but there is another type of subordinate clause that allows for a separate subject clause to the main clause but still functions a descriptive clause and that is the demonstrative clause.

"The lady, who will be holding a rose, is a spy."

Now, through the use of "who" as a demonstrative pronoun, it is possible to give two pieces of information to the listener at once, as there are two transformative verb clauses. However, one clause is given greater significance as it is part of the main clause.

The difference between this example and the participle clause is only in informing the listener that the information being presented to them might not have been knowable to them earlier in the conversation.

The demonstrative clause can also take an entirely separate noun clause rather than the referential pronoun in the previous example.

"The ice-cream that children like the best is vanilla."

Now an entire sentence is acting as a descriptor. In this example the main clause "the ice-cream is vanilla" loses its significance without its subordinate clause "that children like the best" but is still the core information with regards to the sentence's consequences.

2.4.6 Complex and Compound Sentences

A complex sentence (Linguistically) is any sentence that contains a main clause and a subordinate clause. As well as the examples already given there exist other forms that combine a subordinate clause to the main clause.

This can be either a conditional clause (which is a type of dependent clause) or the use of a subordinator (a type of conjunctive adverb).

"If I were to win the lottery then I will give you half the money."

The main clause "I will give you half the money" is dependent upon the conditional subordinate clause "If I were to win the lottery".

"Because I didn't win the money, I won't give you half the money."

Now the subordinator "because" makes the phrase "I didn't win the money" subordinate to

the sentence's main clause. The subordinator also determines how the subordinate sentence relates to other sentence. So conditional clauses such as with "if", "when" or "unless" are still a class of subordinator.

A compound sentence is a sentence that has more than one main clause that is achieved through the use of a coordinating conjunction (such as "and" or "or"). Now any number of clauses can exist in a sentence or be elided to allow a sentence potentially unlimited in length.

"The horse jumped the fence and a man ran after it."

This raises an issue for the idea of similarity because now it would be possible to have another input which has split the sentence into two parts: "The horse jumped over the fence. A man ran after it."

2.4.7 Parts of Speech

The parts of speech and the clauses can normally be determined by the structural words (Quirk, 1962) and morphemes. The core parts of speech can be classified (McArthur (ed.) , 1991) as follows: Articles; Nouns; Pronouns; Verbs; Adjectives; Adverbs; Interjections; Interrogatives; Conjunctions; Prepositions; Auxiliary Verbs; Possessives.

Many words have little semantic meaning on their own but still fix the meanings of the words around them. An obvious example of this is to return to the idea of the hypernym or the "is a" ontological relationship. Both "is" and "a" are normally excluded as stop words by sentence similarity models, yet, these words define the relationship between a pair of nouns that have been used to build WordNet (Feldbaum, (ed.), 1998). These words are sometimes described as structural words (Quirk, 1962).

2.4.8 Direct Objects

Verbs can take direct object clauses to form a predicate. Some verbs can take more than one direct object and are known as ditransitive verbs because they take two objects and verbs which take one object are known as monotransitive verbs (McArthur (ed.), 1991). There is no convenient list of all the ditransitive verbs in English and many words can take more than one object especially when the first object is "you".

"I will walk."

"I will walk the dog."

"I will walk you home."

Verbs can also have prepositional clauses:

"I will give the money to him"

The meaning of verbs can alter with the preposition such as "push down" or "push for" or when it takes an object such as in a predicate clause.

A very small class of verbs can take a direct verb clause without the need for an additional "to".

"Make her do it" as opposed to the more normal structure in "force her to do it" which includes "to".

2.4.9 Implied Words

One of the most complex elements of English for a parser is that patterns can be elided so that key words can be omitted and still leave perfectly valid sentences and increased ambiguity.

The most common implied words are "then" and "that" but other longer phrases can also be omitted. The optionally omitted words are shown here using square brackets.

"If I were hungry [then] I would eat an apple."

"The ice-cream [that] the children liked the best is vanilla."

The combination of implied words and ditransitive clauses can make coping with ambiguity in the object clause challenging for automation.

2.4.10 Perfect Ambiguity

Just as the individual meaning of polysemous words could require context to resolve their meaning the same situation can arise where the structure and hence the meaning of a sentence can be perfectly ambiguous without wider context.

An example of this can be found in the Oxford Companion to the English Language (McArthur (ed.), 1991) under the section on transformational generative grammar:

"Visiting relatives can be a nuisance."

Here, the sentence has two possible configurations of the subject clause: a participle clause with a direct object; or a noun clause with the participle "visiting" function as an adjective. Adding the specificity of a possessive pronoun would cause the structure to again be resolved:

"My visiting relatives" and "visiting my relatives".

In some cases where the parts of speech would be identical there can still be different identifiable clauses.

The following example is closely related to a pair of sentences in the Cambridge Encyclopaedia of Language (Crystal, 1984) again discussing Chomsky's work:

"He is easy to please."

"He is eager to please."

Although, it would be possible to view this pair of sentences as having the same pattern, it is also possible to consider the sentences as different disambiguations of the same possible choice of patterns. The verb "please" is actually used in the first case as taking a direct object in a demonstrative sentence. It could be re-written as "it is easy to please him." There would be a change in meaning though were the same transformation done to the second sentence: "it is eager to please him."

Despite having the same possible pattern, a human will disambiguate the sentences' structure differently because of the different meanings between the adjectives.

2.5 Automatic Parsers

Computational Parsing has been around since the 1960s with the Brown's Corpus being an early tagged corpus annotated for part of speech (Francis and Kucera, 1979). Other tagged datasets and automatic parsers have developed since. The Penn treebank (Marcus et al., 1993) has become a standard annotation for modern part of speech taggers. It also includes skeletal clause tagging.

There are a large number of parsers that have been proposed in the literature. Many of which are from several decades back and several openly available. Parsers can basically be classed as either deterministic, probabilistic or a hybrid.

High success rates were found with early part of speech parsers like the Bell parser using just trigrams with lexical and context probabilities (Church, 1988). Later probabilistic parsers use learning methods such as Hidden Markov Methods (Brants, 2000), and

maximum entropy (Tsuruoka et al., 2005).

While even stochastic methods need some consideration of clauses not all parsers directly output clause tagging. Another area, text chunking, can take the part of speech tags and construct the clause tagging (Federici et al., 1996) and are included with the Natural Language Tool Kit (NLTK, 2013). Several parsers will output the clauses or parse tree such as CLAWS (Garside and Smith, 1997).

Most probabilistic parsers are supervised or semi-supervised (Erk and Pado, 2006) some aim to find rules without prior knowledge and identify the patterns within the corpus (Chrupala et al., 2008), (Spoustová et al., 2009). Some are trained on specialist medical datasets but can still be used in the general case (Tsuruoka, et al. 2005)(Denis and Sagot, 2009).

Deterministic parsers use a set fixed Linguistic rules in order to makes a decision. They aim to chose from a set of valid patterns for the sequences of the words and can require arriving at the end of an expression before reaching a decision and are generally slower than probabilistic methods. The Brill parser (Brill, 1992) is an early example of a rules based parser. Currently available rule based parsers include MaltParser (Nivre, 2003), MSTParser (McDonald et al., 2005) and RDRPosTagger (Nguyen et al., 2011).

Parsers have achieved a very high match for human judgement of correct parsing of above 97% accuracy on complex datasets such as the Stanford POS Tagger (Toutanova and Manning, 2000) and continues to develop by adding more Linguistic rules for the Stanford 2.0 tagger (Manning, 2011).

Parsers can achieve close to expert human accuracy and include important frequency information but there will always remain ambiguous situations, as mentioned in the previous section on grammar. Parsers can be computationally expensive but with improvements in computing this has become less of an issue.

2.6 Knowledge Sources

In order for a machine (or a person) to interpret the meanings from language, it is necessary to have a source of knowledge that can be used to give the relationships between a fixed set of ideas. There are several approaches that have been adopted in order to obtain meanings or their comparable structural representations for a large vocabulary such as is needed for processing natural language. Ultimately all approaches for similarity require a human source of knowledge which can be one of the following sources:

- Direct human encoding to build a structural representation for each meaning such as an ontology.
- Structured knowledge for humans - such as dictionaries (Lesk, 1986), thesauruses (Kennedy and Szpakowicz, 2008), the semantic web (Hliaoutakis et al., 2006) and encyclopedias (especially Wikipedia) (Gabrilovich and Markovitch, 2007) .
- Examples of usage - these can again include human intervention for supervised learning of tagged language; corpora (Laudnauer et al., 1997) with large samples of real use of English or summary information of frequency of usage such as web statistics like Google Ngrams (Google Ngram website, 2013) .

2.7 Representations of Meaning and Similarity

Language is a powerful abstraction for ideas and there is no definitive mechanism for converting these ideas into something that a computer could understand or use. The key starting point for language is storing the meaning of a fixed vocabulary of words. The approaches for storing the meaning of words for comparison fall into 3 categories:

- 1) *Conceptual architecture* - The underlying meaning of each word is represented by a set of attributes that are connected to the meaning (Pascual and Tunez, 2010a).

2) *Ontology* - Each word is connected to other words using a pointer to represent a specific relationship. Such as "has a" so a "table has a leg," provides a link for a meaning of the word "table" and the word "leg".

3) *Corpus* - The usage of words in their context is used to determine their meanings. The co-occurrence of words is used to identify meanings and form connections. This can either be simply the frequency information or can be used to construct a set of fixed links to represent the meaning of a word based upon its usage.

These then lead to groups of meanings (each represented by an id) and weighted connections between the groups.

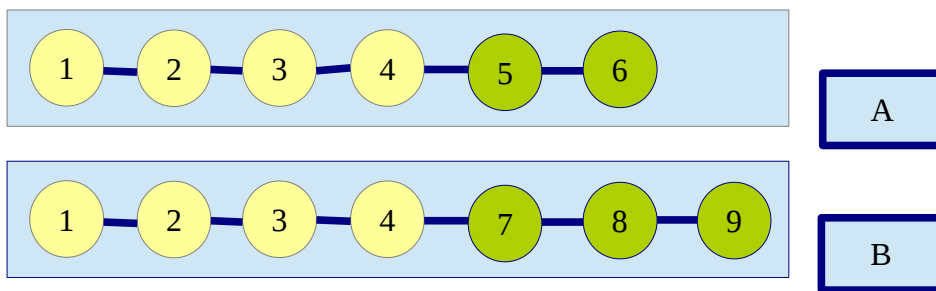
2.7.1 Comparing Structures

The possible structures of the word meaning can be described in terms of standard search spaces. Common techniques involve trying to find the shortest connecting path between two meanings or finding the common overlapping structure.

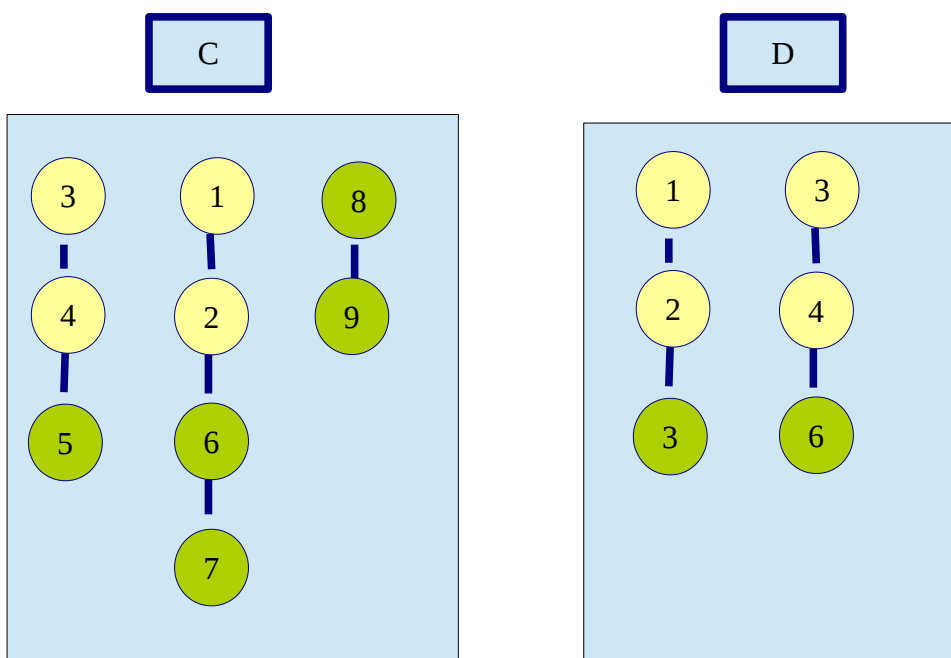
An easily compared source of data is where every meaning has an identical structure with a value for each of a set of parameters and then a simple function call or matrix operation can be defined to give a similarity.

The more common approach is when there is not a single fixed size of structure for every meaning but instead the structure can be viewed as a set of nodes (each meaning of a word being a node) with connecting links forming a web of all the vocabulary.

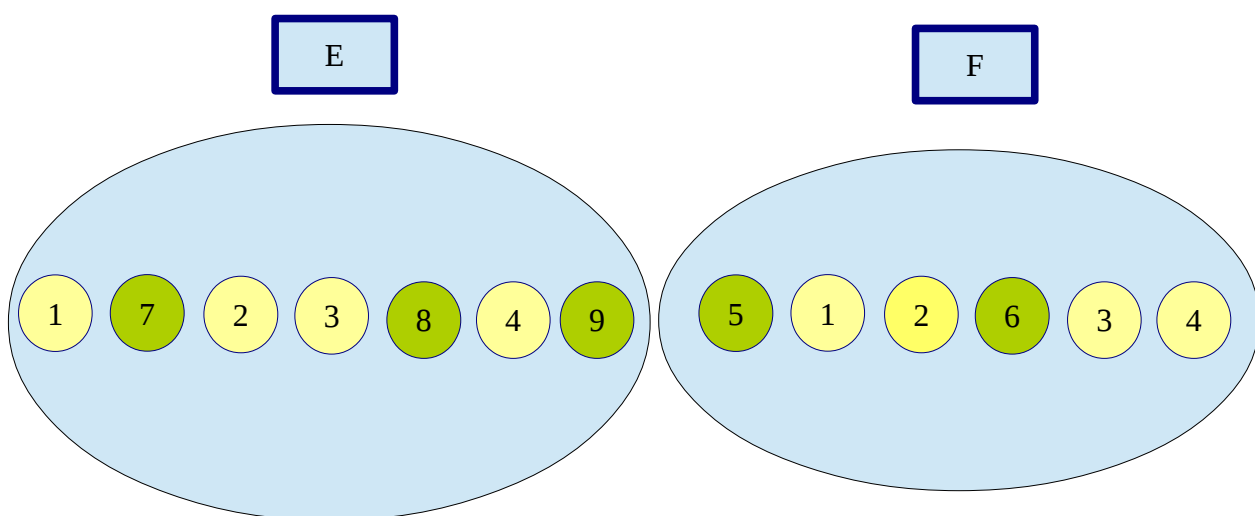
A chain is the easiest variable structure to describe and process as there is a single path to traverse and a single parent for each node until arriving at a root node. Where all chains end at a common level the common structure is simply another chain iterating from the root until reaching the lowest common parent.



Chains A & B



Multiple chains C& D



Sets of properties E & F

Figure 2.1: Possible meaning structures pairs

Figure 2.1 shows three pairs of meaning structures that could be commonly need to be compared in order to judge the similarity. In each example the circle with a number in it represents an individual node and the overlapping structure in each case is given by 1, 2, 3, 4. The top example A & B shows two chains with a common root node. C & D is the case where each meaning now has multiple possible chains. The order of the chains is not important. The final example is where there are no connections between the nodes and they are stored as ordered sets. These structures can be processed as follows:

- *Two chains* - Where there is a common root level then it is only necessary to iterate over the chains from the root node until the values do not equal one another. This is the lightest algorithm requiring at most the same number of steps as nodes in the shorter chain.
- *Multiple chains* – This would be a situation where there are a limited number of root nodes and then the chains are first compared using the ordered set followed by a chain comparison.
- *Sets of properties* - A pointer is used for the current position in each set of properties. If value 'x' is less than value 'y' then 'x' is advanced to the next property, else if 'x' = 'y' then this means a matching property, otherwise 'y' is advanced to the next property. The number of comparisons needed to find the overlap for two unordered sets of size 'm' and 'n' (where 'n' <= 'm') is $m * n$. However, in most cases the sets can be ordered in which case the largest number of comparisons is $(m + 2 * n - 1)$.
- *Two nodes in a web* – In this case only one structure and the objective here is normally to find the shortest path between the nodes. Since it is possible to have to go in both directions along a link, the search space can be very large.

As a vocabulary is fixed, for the slower methods it is sometimes the case that the results will be stored in a look-up rather than used for direct calculation each time (Tsatsaronis et al., 2010).

2.7.2 Conceptual Descriptions

There are approaches to construct a universal description of terms based upon how language is deemed to be understood by people (Kaschak and Glenburg, 2000) with attempts to conceptualise the language as relating to mental models (Zwaan, 1999).

Context free grammars that can be used for processing both machine and natural languages have been suggested where sets of elements and rules are combined together (Knuth, 1964). Other constructions have been proposed such as functional grammar (Dik, 1997) which focuses upon functions within language deemed to be fundamental to its understanding. Later these constructions have been extended to functional discourse grammar using the spoken form of language as the key component (Hengeveld and Mackenzie, 2008).

A functional grammar architecture has been used for a project called FunGramKB (Functional Grammar Knowledge Base) which has built an architecture combining a lexicon, a grammicon, and a conceptual framework (Pascual and Tunez, 2010a). The conceptual framework is designed to be universal to cover an ontology, procedural knowledge and knowledge connecting events (Pascual and Tunez, 2010b). It is not a complete dataset but an ongoing work with many conceptual elements and so has not been directly used for sentence similarity as its main focus is for NLP (Natural Language Processing) applications giving an understanding of the meaning.

2.7.3 Ontologies

An ontology uses a set of relationships in order to define the connections between meanings.

The simplest relationship is the synonym. Here there is more than one form of a word that shares the same meaning. This means that the words could be used in the same context and convey the same meaning to the recipient.

The most important relationship to comparing meanings is the hypernym relationship. This is given by an "is a" or "is a type of" link. For example, "an elephant is a mammal." A mammal is also an animal. Of course, an elephant is therefore an animal. By storing the direct hypernym for each word, this then allows for a chain of hypernyms to be formed down to a root node.

The reason why this becomes important to similarity is because each word has only a single hypernym parent. If we were to compare "cat" to "elephant" then because both share a common hypernym of "mammal", it is known that the word "mammal" represents a common idea between the two words.

Other ontological relationships exist such as the meronym, where one word is part of the other, which can be expressed by "has a". So the word "finger" is a meronym of the word "hand" because a hand *has* fingers,

Another commonly found ontological relationship is that of antonyms where a pair of meanings are directly opposite to each other on a particular scale. So "good" and "evil" are antonyms.

A fifth relationship that is important to the Linguistic meanings of words is the "is" relationship. This defines a word that is an attribute or a property of another word. So an elephant *is* grey. Where "grey" becomes a property of the word "elephant".

2.7.4 Sentence Relationships

Just as ontologies describe the relationships between known words, other similar logical relationships can exist at the clause and sentence level. These can be used for comparing the relatedness of sentences rather than the semantic similarity. Some relevant ideas for sentence similarity are:

- *Paraphrases* - a pair of sentences with the same meaning expressed in different words equivalent to synonymy.
- *Contradiction* - when there is an assumed common context that two statements can't be true at the same time.

"John Smith alone killed the president's dog."

"Michael Jones alone killed the president's dog."

If it is assumed that the same president's dog is referred to in both sentences then both sentences cannot be true. There are also weak contradictions when two sentences can both be true but are not very similar.

- *Opposites* - when two sentences share a common scale, then there can be two diametrically opposite sentences such as was the case with antonymy. There are 3 core types of opposites: negatives (" I love you " / "I don't love you"); antonym verb clauses ("I hate you" / "I love you"); and inversion of the subject and object for some verbs ("The blue team beat the red team" / "The red team beat the blue team").
- *Question and Answer pairs* - this final relationship is often encountered with respect to information retrieval (Baeza-Yates and Ribeiro-Neto, 1999). Two sentences can be semantically very different but can be useful to finding answers to user's requests. This creates a choice from a set of possible answers can made based upon an input question.

2.7.5 WordNet

WordNet (Feldbaum (ed.), 1998) is a widely used and respected ontological database that was developed at Princeton university (Morato et al., 2004).

WordNet encapsulates a large body of knowledge into a single source and provides a common set of relationships that can be used between the terms it contains. There are issues of consistency as to how the relationships have been encoded for various terms.

It fails to distinguish between the various updates in its vocabulary via its version number. This means that it is possible for two people to use WordNet with the same version number but be using slightly different ontological relationships.

The online version has remained unchanged in recent years but there are implementations such as with the Natural Language Toolkit (NLTK: Natural Language Toolkit website, 2013) which differ in terms of some of the vocabulary. The meanings are associated with the stem of a word which means that any word has to be stemmed prior to being looked-up within the database. Structural words are not included inside of the ontology. Only the four main word types of nouns, verbs, adjectives and adverbs are included.

The meanings are grouped into synonym groups with a dictionary definition for each meaning and ontological pointers between each of the synonym groups. The nouns have the richest ontological structure with the hypernyms being the most extensive ontological relationship.

is a valuable resource that encodes a large amount of human knowledge. Its creation enables many sentence similarity projects to operate using real inputs that otherwise would likely have to have been constrained to a limited vocabulary. It contains a large number of obscure meanings that have the same form as common words so represents the high complexity that results for automation from disambiguation, which unlike most tasks can become more challenging with greater knowledge.

There are many places with a lack of consistency in its architecture and so in places there is a distinct variation in how meanings are encoded which probably should not be present. As a result the dataset can be fairly noisy in places when considered for its vocabulary and ontological relationship.

There are a few instances where the inconsistency even deviates from the design specifications. Such as where a word has been given two parent nodes with a hypernym

relationship or when an irregular form is defined in its database but in a manner that cannot be found internally.

The data is inefficiently processed internally where it is not using the fact that it is alphabetical and the structure not built for easy expansion. However, the database provides a solid approximation of the relationships between meanings and enables the implementation of sentence similarity models using it as a knowledge source. It also is the standard choice for knowledge based sentence similarity models (section 2.10) and allows for a consistent comparison from an academic perspective between models too.

2.8 Word Similarity Measures

Word similarity is a way of producing the similarity between a pair of words that are being compared without the context of a sentence. Without further context, the highest overlapping meanings between polysemous words are most likely to be assumed by a person without other guidance as to which meanings to use.

While it would be possible to simply encode similarity ratings for every possible pair of words in a vocabulary, this is impractical. Instead methods use the meaning representations (discussed in section 2.7) to give an expression of the similarity between two meaning representations' structures (see section 2.7.1).

While there are differences between the performance of the algorithms with regard to how the structures combine compared to human judgement of similarity, it is predominantly the meaning representation that dominates the accuracy of a particular model.

From a Linguistic point of view the word similarity methods, once the meaning structures have been compared, are about similarity not the meaning of the word. Hence, the only requirement from the Linguistic perspective is that a model provides a solid approximation of the similarity that could be used with a sentence similarity model.

There have been numerous word similarity measures proposed, some of which use the ontological connections within a hypernym chain structure with each node having one immediate parent. These either take the lowest common parent (a.k.a. lowest common subsumer, LCS, or hypernym, LCH) as a node or use the depth of the node. There have been a range of functions to combine these parameters proposed.

Wu and Palmer (1994) take the double depth of the LCH and divide it by the sum of the individual word depths. Leacock and Chodorow (1998) use the LCH depth divided by the maximum possible depth inside the structure. Li et al. (2003) examine a wide range of possible word metrics for performance with WordNet using the total distance between two meaning nodes and the LCH. There are several other more recent variations and weighted functions using the same parameters and these can be normalised to a similarity scale of 0 to 1 by dividing by the maximum similarity within the knowledge base (Mihalcea et al., 2006)

Other word similarity measures are based upon the frequency of occurrence within the corpus of either the words co-occurring or of the hypernym word. Point Mutual Information (PMI-IR) takes a pair of words and looks for their co-occurrence (or more refined occurrence in close proximity) and divides this by the total occurrence of the individual terms and then takes a logarithm in order to give an overall similarity (Turney, 2001). Resnik (1995) uses a simpler metric of the frequency (probability of occurrence) of the LCH and assigns an information weight of minus the natural logarithm. Lin (1998) further extends this to include the Information weights for both the individual words too.

When comparing polysemous words without context, normally the highest similarity score is used, however, sometimes disambiguation can also be included. Lesk (1986) proposed a method of word meaning disambiguation when comparing words by comparing dictionary definitions and this has been used with WordNet (Banerjee and Pedersen, 2002).

Some of the parametrised and more recent proposals for word similarity models have shown some issues of over-tuning where parameters were tuned specifically to give optimal values for the test dataset as opposed to the general case (Zesch and Gurevych, 2010).

Also of interest for using distinct structures are OMIOTIS's word model (Tsatsaronis et al., 2010) and PoW (Properties of Words) (Pearce et al., 2011).

OMIOTIS includes a word similarity measure using the shortest path length between two meanings in WordNet using all of the ontological relationships stored and so has a mesh / web structure to navigate.

PoW requires sets of properties to represent the meaning of each word, and although a proposal for semi-automation from knowledge sources such as from a dictionary (Pearce et al., 2011), it would require human knowledge in order to construct these meaning structures. It simply combines a ratio function using the common properties and distinct properties of each meaning.

Other word similarity models have been incorporated as part of sentence similarity methods discussed next.

2.9 Corpus Based Sentence Similarity

Corpus methods use the co-occurrence of words in large corpora in order to make judgements as to how similar meanings are, ranging from words to documents in size. The more sophisticated methods use a corpus to build semantic vector spaces to represent the meanings of words, thus making it possible to compare words with similar meaning even if they don't appear in the same document.

Two of the best known approaches are Latent Semantic Analysis, LSA (Deerwester et al., 1990), and Hyperspace Analogues to Language HAL (Burgess et al., 1998). It was shown that HAL was not as effective as LSA for small texts due in part to sparseness (under 200 words) so is less significant for the purposes of sentence similarity.

- **LSA** (Deerwester et al., 1990) is perhaps the most significant corpus method for the field of sentence similarity. While primarily designed for comparing large texts, it performs strongly on short-texts. LSA aims to find the underlying "latent" connections between words. A matrix of the terms-by-documents for each term has its dimensions reduced through removing sparse features using singular value decomposition. LSA then takes a cosine between two formed vectors to create a similarity score. LSA has no contribution for the effect of word order or punctuation (Li et al. 2006).
- **ESA** (Explicit Semantic Analysis) (Gabrilovich and Markovitch, 2007) is built using Wikipedia to build a representation of words based upon their frequency of occurrence within articles being used as contexts. The vector for a word is found based upon a set of weighted concepts.

Wikipedia articles can contain user defined hypertext links to other related Wikipedia articles. Synonymous terms can be identified as the displayed word for the hypertext link and need not be the same as the linked article name. Each article then can be converted to a set of concepts from these links in order to give a set of terms that relate to the article title.

As with latent semantic similarity (LSA) these terms are then weighted and can be used to construct a vector of concepts. These links can be further used to disambiguate the meanings of terms that could link into more than one possible article. Whereas latent semantic analysis supposedly aims to extract a set of concepts from each term based upon the occurrences in large corpora, ESA uses the human structured information from the encyclopaedia to extract its concepts.

A reduced set of the English Wikipedia was obtained through removing articles with fewer than 5 links within the article or to the article from another article. Stop words (common structural words) and rarer terms were extracted to leave them with about 390, 000 terms after stemming from 240,000 articles. These terms or concepts can then be used to create weighted vectors for two inputs which can be sentence length or larger documents. The vectors are combined using a cosine.

ESA is primarily focused on relatedness of words as opposed to being specifically focused on meanings of words. This means that rather than relying on structures relating to the meanings of the words that it also includes associations.

- **SSA** (Salient Semantic Analysis) (Hassan, 2011) is closely related to ESA which is also using Wikipedia as a corpus but with a more refined mechanism for handling "anchor words" which are synonyms occurring within the articles.

SSA extended the source of articles being used from ESA so that foreign language versions of Wikipedia in Arabic, Romanian and Spanish in addition to the English Wikipedia were included. Unlike ESA, SSA uses the surface information provided by the hypertext links as opposed to simply the keywords. This then allowed for the connection between a term taken as the article's title and the documents to which the titled article is linked to be combined,

SSA is very similar in scope to ESA and is focused on associations as well as other potential ontological relationships. It produces a weighted vector for each input being compared which it converts to a single value using a cosine.

- **IISIS** (Islam and Inkpen, 2008) produced a sentence similarity model that uses a search engine corpus (Alta Vista) and in addition to the corpus techniques has also included word order as introduced to sentence similarity by STASIS (Li et al., 2006). Rather than adding this as a separate similarity as is the case with STASIS, the order similarity is the primary comparison. This is achieved since for two locations of words to be compared, there must be an overlapping similarity between the words. As is the case with the other corpus methods, it removes all auxiliary verbs and structural words and lemmatises the words to their stems. Additionally IISIS also includes a string matching part to its algorithm allowing for closely written words to be found.

2.10 Knowledge Based Sentence Similarity

In contrast to the corpus based methods, other models take a human defined lexicon including ontological relationships to obtain a structure (such as in section 2.7.1) that can be used to compare a pair of meanings. This source of chosen knowledge that has chosen has almost always been WordNet (Feldbaum (ed.), 1998) (discussed in section 2.7.5).

- **STASIS** (Li et al., 2006) measures the similarity of pairs of forms using only the nouns within WordNet. STASIS stems the words and combines all the words into a bag of words to obtain equal length vectors. It includes the Li et al. (2003) word model with Resnik (1995) information weights. It also introduced a word order comparison as a separate vector.

Each word in the bag of words is compared against each word in the other sentence and the highest similarity chosen for each word. This gives a vector the same length as the bag of words.

A second vector is formed based upon the ordinal value of the word with which the highest match was made, provided that the similarity between the pair of words is above the threshold of 0.2.

Finally, the two vectors are combined using the second order normal which can be used because the vectors have the same length. This gives similarity scores for both the meaning and for the word order similarity which are then combined using a ratio of 0.85:0.15 for the meaning to word order similarities.

- **DTW** (Sentence similarity with dynamic time warping) (Liu et al., 2007) - the key feature of this method is that the similarities between the words are used to form a matrix which then uses an alternative method of reducing the similarities to a vector using dynamic time warping.

Dynamic time warping is a method that can be used to compare two sequences and create a matrix between the points on each sets to create a minimum distance through the matrix. This approach has been used by putting the word similarity scores using the Li et al. (2003) word algorithm between the keywords of a sentence to get a single value.

With DTW the matrix of numbers that results from making all of the possible word comparisons is given a single path through the matrix. This could be thought of as selecting the closest path to the meaning, before combining this value into a single value.

- **OMIOTIS** (Tsatsaronis et al., 2010) like STASIS interacts with WordNet but rather than restricting itself to the noun hypernym chains, all of the WordNet ontological relationships are used for all the words. This means that finding the shortest path between words requires a more complex search. To compensate for this additional computational expense all possible word pairings have had their similarity stored to be retrieved via look-up. For sentence similarity, importance weights are added and the harmonic mean is used to give the overall similarity measure.
- **SyMSS** (Oliva et al, 2011) - a more recent model where the parse tree structure has been added to sentence similarity models and included as an additional factor to the comparison.

SyMSS in essence combines the parse tree with a variety of the extant word similarity models using the vocabulary from WordNet. It combines the overlapping structure and makes a subtraction from the structures that are non-overlapping between the two experiments. Finally it combines the values from the word similarity scores with the common structure of the parse trees to obtain a single similarity score.

- **FAST** (Chandran et al., 2013) an extension of STASIS to include fuzzy knowledge based upon scaled sets of six dimensions such as size, effectively improving the comparison of adjuncts.

Other approaches have extended some of the main models either with different word models (Achananuparp et al., 2008) or knowledge bases such as using a thesaurus (Kennedy and Szpakowicz, 2008). The models all share common features in that the word interaction is largely not included beyond the order of the words with the lemmas often being used with structural words such as "not" and "the" being completely excluded.

Interesting work is also being performed by Mitchell and Lapata (2010) although they have been concentrating on simpler units of language beneath sentence length. It uses a vector based approach rather than closely adopting Linguistic concepts but does introduce some level of extensibility.

2.11 Relatedness Measures

There are other tasks which closely relate to semantic similarity which instead compare the logical relationships between parts of sentences. There are a large number of models which can compare sentences. Such as: general relatedness similarity measures - such as CHESA (Lieberman and Markovitch, 2010); or sentence similarity focusing on differences (Qiu et al., 2008); or a hybrid (Ho et al., 2010). Other models are designed to find specialist relationships such as question and answer pairs (Gaizauskas et al., 2005), and paraphrases combining the matrix of meaning overlap (Socher et al., 2011) and machine translators (Madnani et al., 2012).

2.12 Limitation of Current Sentence Similarity Models

When examining the existing sentence similarity models described in the previous two sections from a Linguistic perspective using the principles identified in sections 2.1 and 2.3, it can be seen that the existing models cannot handle many features of English.

By taking lemmas or stems of words some of the structural information from words is automatically being ignored. In some instances, this will be information directly affecting the meaning, such as being a plural. In others, it is additional structural information for

how the words are combining together to form more complex meanings that is lost.

All the ontological methods, even when being restricted to just nouns, still often will have to decide what meaning to use for polysemous words. The only basis for this decision comes from the comparison between sentences and not the words used in the sentence itself.

Corpus methods (LSA, ESA, SSA and IISIS) have a level of context included from the probability of co-occurrence of the words. The problem is that they will pick the most likely meaning based upon the general case but not necessarily consider the intended meaning in its context if it is a less common usage.

Word interaction is not fully handled by any sentence similarity model and in most cases the structural words are being completely excluded on the basis that they hold less semantic information. This includes words which can have significant effect on the meaning such as “if” and “not”. Likewise the temporal effect of auxiliary verbs is also potentially excluded.

While STASIS (Li et al., 2006) has introduced a word order similarity component to sentence similarity, the word interaction cannot always be represented by the word order. When for example there is a prepositional clause, the word order is altered from the inclusion of extra words. Similarly, it is the function of the clauses, not the order of the words that matters to the meaning.

Even SyMSS which includes the parsing information still does not combine this information with the words.

As a result of not adopting a Linguistic approach, no sentence similarity model is currently using the word interaction of how the meanings combine and most exclude the information that adds this information were it wanted to improve the algorithms. Similarly no model tries to specifically determine, the meaning of words as a human would in the sentence.

The models are very much topic focused as opposed to being focused on the semantic similarity of the sentence. This means that there are many situations within English that are

not currently handled by any semantic sentence similarity model and why this research is pursuing a Linguistic approach to aim to improve the accuracy of sentence similarity.

Finally, as will be examined in chapter 14, no consideration has been given to the logical relationship of opposites by any sentence similarity model which can be very significant to similarity.

2.13 Conclusions

This chapter started by presenting Linguistic concepts with a focus on how it could be applied to comparing the meanings of sentences. Several key concepts were identified including topic, word interaction and context. These concepts will be used in chapter 4 as the basis of the Linguistic framework that is critical to this research and its objective (section 1.4).

A discussion was given on how the structures of English can be identified and how this has been automated with parsers. Parsers are an important component of a sentence similarity model as they can add the Linguistic information needed to compare sentences using a Linguistic approach.

The chapter then presented how the comparison of meaning for sentence similarity has developed, starting from word comparison and ending with the latest ontological and corpus based similarity models.

The final step was an examination of how the Linguistic concepts, presented at the start of the chapter, relate to the limitations of the pre-existing approaches to semantic sentence similarity. The next chapter presents the datasets needed and used as part of the evaluation of the research into sentence similarity with a Linguistic focus.

The final analysis of the existing sentence similarity models shows how there are several

elements of English that have been identified with Linguistics which are not being handled. This highlights the points made in the motivation of the research in section 1.5.

3.0 Datasets

3.1 Introduction

The objective of this research, set out in the introduction, was to investigate whether a sentence similarity model could be improved via the inclusion of Linguistic concepts. Limitations in the standard datasets (discussed 3.2.1) showed that these did not contain sufficient variation to evaluate a sentence similarity model, particularly when using a Linguistic approach. Therefore, there was the need to expand on the existing datasets to allow for direct evaluation of the sentence similarity models being presented in this thesis.

This chapter gives an overview of the key pre-existing sentence similarity datasets used as part of the experiments in later chapters, alongside some of the key word similarity sets that closely relate.

The methodology and approach used for the creation of the new datasets in this research is given before presenting the three new datasets needed for the experiments (outlined in chapter 5). The ten pairs dataset and its extension, the thirty pairs dataset, follow the same approach as the standard dataset (the STASIS-30 dataset (O'Shea et al., 2008)) used in the literature. However, the opposites datasets uses a new scale for sentence similarity.

This new scale is a significant step for sentence similarity and the opposites dataset can be regarded as a specialist domain. This is a very significant step for sentence similarity but the idea is mainly presented in the final stage of development in this thesis in chapter 14.

3.2 Words Datasets

Words represent the simplest form for comparing the similarity between meanings. Section 2.8 gave an overview of the methods that have been used to compare the similarity of words. Many of the knowledge based sentence similarity models discussed in section 2.10 use word similarity measures (section 2.9) that have been tested using the standard word similarity datasets which are given below.

The sentence similarity models being developed as part of this research do not directly use the word pairs dataset, although components of the word similarity module first presented in the next chapter have been evaluated on them.

Rubenstein & Goodenough (1965)

The standard word pairs 65 pairs of nouns.

Miller And Charles (1991)

This is a subset of thirty of Rubenstein and Goodenough pairs which are regarded to give a better balance of similarity. Taking the pairs originally numbered 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 47-65 in the Rubenstein and Goodenough dataset.

A slightly more sophisticated dataset finds the similarity between pairs of word pairs which is a level between sentences and word pairs. This dataset was important for evaluating the impact of changing the underlying word similarity formula:

Mitchell and Lapata (2010)

Pairs of coupled words with fixed part of speech. 3 sets of the most frequently occurring: Noun-Noun; Adjective-Noun and Verb-Noun. Each set is 108 word pairs rated for high, medium and low similarity.

3.3 STASIS-30 Dataset

The STASIS-30 dataset (O'Shea et al., 2008) has become the standard dataset for benchmarking sentence similarity and is widely used in the literature. First used with STASIS (Li et al., 2006) hence the name. However, as all of the sentences are definitions, which include the original keyword within the sentences, the Linguistic variation in the dataset is limited. With one exception, every single sentence has the verb "to be" as the main verb.

The single sentence definitions for the Miller and Charles (1991) nouns were selected from the Collins co-build dictionary (Sinclair (ed.), 2001). This then produces pairs of sentences which were rated by 37 participants to give similarity scores which were collated using the means of the values. The use of solely definitions has limitations with respect to its use for evaluation of the sentence similarity models including Linguistic features, discussed next.

3.3.1 Limitations of STASIS-30 Dataset

The sentences from the definitions are effectively tautological for the head word being defined (which is also contained in the sentence). These defined words have also had their definition selected based upon the original word comparison in the Miller and Charles (1991) dataset. This leads to the situation where the closest possible meanings between the words is likely being picked, which would be the same case as with the underlying word model. This strongly favours the selection of meanings of the words based upon the compared sentence rather than the individual context of a sentence.

More significant still is the lack of variation in the verb clause. All of the main verbs are "is" so the sentences are purely descriptive not transformational (section 2.4.2) and there is extremely limited word interaction as a result and the variation in Linguistic structure between the sentences is artificially lowered. The standard dataset (STASIS-30 (O'Shea et al., 2008)) therefore is very sparse in terms of word interaction and context which are critical for the purpose of this investigation.

3.4 Microsoft Research Paraphrases Dataset (MSRP)

The MSRP (Microsoft Research Paraphrase dataset (Dolan et al., 2004)) is the largest extant set of human classified sentence pairs. The sentences were originally taken from news articles which were sourced from the AP (Associated Press). The pairs are selected from different articles which were created from the same story and so should be conveying the same information in both cases.

Each sentence pair is rated as to whether or not it is a paraphrase. Three people were used as the raters. Each sentence pair is first rated by two of the raters and where there was disagreement in the rating, the pair of sentences were shown to the third rater to decide the classification. The total set comprises 5801 sentence pairs randomly divided into a test set of 1725 pairs and 4076 pairs which could be used as training set (Dolan et al., 2005).

3.4.1 Limitations of MSRP for Sentence Similarity

While the MSRP represents a very large source of rated sentences, there are issues in that the paraphrase relationship is not the same as sentence similarity, which is a point discussed in more detail in chapter 13. The main issue is that the dataset is a specialist dataset consisting of highly similar meanings between the sentences that have not been rated for their similarity score.

This dataset will be regarded as a specific specialist domain for use later with the developed sentence similarity model. It however does not address the shortfalls needed to be resolved for the use of the evaluation of a sentence similarity model and its performance from the inclusion of Linguistic concepts. Therefore, there was the need to create some new datasets.

3.5 New Datasets

It was shown how the selection of the sentence pairs for the STASIS-30 dataset has issues for testing sentence similarity models (section 3.3.1), in particular when using a Linguistic focus. Equally, completely randomly selecting sentences would not be likely to yield a suitable test set as there would be many very weakly or unconnected sentence pairs.

This means that in order to assess whether the objective of the research, to see if the inclusion of Linguistic concepts, could improve a sentence similarity model some small new datasets were created called: the ten pairs dataset; the thirty pairs dataset; and the opposites dataset. This chapter provides details of the approach used to create the new datasets needed as part of the evaluation of the research.

3.5.1 Design Approach and Constraints

This section gives an overview of the design approach and constraints for producing rated sentences for new datasets. While any pair of written English sentences that have been rated for their similarity score using human judgement, represent valid input for directly testing a sentence similarity model's performance, some consideration needs to be given to the characteristics of the dataset.

Two types of dataset were created to aid the evaluation of the sentence similarity models produced as part of the investigation into the inclusion of Linguistic concepts to a sentence similarity model. Firstly, a general purpose dataset was created, looking to contain variation in the core fundamental concepts that relate to sentence similarity outlined in section 2.3. The other dataset is a specialist dataset designed to focus on testing sentence similarity with respect to opposites.

A comprehensive dataset would need to include every single combination of distinct classifications of English sentences. This is not feasible due to the vast number of combinations that would result. For example, Strang's identification of even just the modal and temporal forms of the verb clause (Strang, 1963) in a basic sentence (subject - verb - object) would result in over a quarter of a million combinations without altering the vocabulary of the noun clauses or the verb. Therefore, the only consideration with regards

to comprehensiveness can be for minimum inclusion of a feature.

A representative dataset would require that the features being tested occur at the same frequency as would be found in English as whole or in a specific domain for which the sentence similarity model were intended to be used. However, for testing purposes, a dataset does not need to be representative. It is only important that the relative performance between two models can be judged based upon their performance on the dataset and that the dataset must contain the sufficient variation in order to test the particular concepts under investigation at each stage. Equally, complete random selection of sentences would not be likely to yield a suitable test set as there would be many very weakly or unconnected sentence pairs.

The contribution from every factor that affects the semantic similarity of a pair of sentences can be thought of as having a "signal". It is the detection of this potential signal from individual factors that is of interest to the experiments. The performance metric (such as correlation) of any particular sentence similarity model on a dataset will also be affected by "noise".

A common Linguistic feature could be affecting the similarity scores for each sentence pair but with its contribution being comparatively small to each sentence pair. This could potentially make it challenging to distinguish its signal from the noise on the sentence. The selection of sentence pairs where the contribution has a larger impact than would normally be the case, would then mean that the signal for the dataset can be strengthened making for easier evaluation. Therefore, no effort is made for representativeness beyond that the relevant ideas appear in the dataset.

The noise is also greatly reduced through the experimental approach (detailed in chapter 5). The core experiment looks at the relative performance between versions of the model with just one factor added. This means that much of the noise would be the same for both models. In conjunction with the not needing to tune any parameters or learn from the dataset, it is possible to use much smaller datasets from there being less issues from the noise.

Finally, after creating the sentence pairs, these need to be rated. In terms of guaranteeing independence, using a third party (distinct from experimenters and raters) to create the

sample sentence pairs could be preferable to direct creation by the experimenter. In the case of this research, however, the design and implementation of the sentence similarity model does not depend upon the dataset but the use of rules based upon Linguistic principles and tunes no parameters with the dataset and is therefore independent of the dataset.

Unavoidably, the generation of a dataset by a third party to meet the desired requirements of the Linguistic variation within the dataset would have still introduced significant influence via the needed instructions and via a final selection of the pairs to be included. It is common practice within Linguistics to create pairs of sentences as illustration of a particular concept and therefore there are minimal issues with using direct creation for creating new datasets and this approach was used here.

3.5.2 Method for Generating Pairs

There was no formal methodology for the construction of the datasets, but a general approach to give the desired variation in Linguistic features within the sentence pairs used to form a dataset.

The objective was to create the datasets that would be needed to evaluate the sentence similarity model with respect to the objective (section 1.4) and the experimental approach outlined in chapter 5. The creation of the sentence pairs can be thought of as involving the following 5 stages:

- (1) Identify the key concepts that were important to include variation of within the dataset.
- (2) It is also wanted for the dataset to be computationally non-trivial, without being difficult for a person to interpret the meaning or the ideas that need to be compared. This involved identifying areas of complexity which should be introduced to the dataset to avoid masking the influence of a factor being included from co-incidentally favourable performance from the algorithm which might not apply in the wider context.

- (3) Create or select sentences, consciously selecting the connection between the pair of sentences. This was done keeping steps (1) and (2) in mind but without the connection specifically matching any inclusion requirement in most cases.
- (4) Select from the pairs created in (3) ensuring that the minimum inclusion criteria are met so that the concepts being tested are at least present in the dataset.
- (5) Show the set of sentences to a group of people to rate for their judgement as to the similarity score.

Steps (1) and (2) did not involve creating a specific list of parameters that had to be included but simply were designed to give an overview to highlight likely problem areas that would occur if just blindly jumping into step (3).

The creation of the sentence pairs would have required a connection even if not identified at the point of creation. This means that even selecting pairs of sentences from random sources would still not be independent from their creation. The creation of sentence pairs as examples in Linguistics is common practice and while step (3) was less formal a connection than a Linguistic exemplar, it still fundamentally is perfectly valid to create pairs of sentences in this manner. As long as the rating of the sentences is independent then the similarity rating can be considered independent.

3.5.3 Human Ratings for Similarity Scores

The use of human judgement is the best available method for approximating the absolute similarity of a pair of sentences. The standard approach of using human rated scores based upon their opinion of the similarity for each sentence pair, then combining the scores as their mean was adopted to implement step (5) of the method for the new datasets being created as part of this thesis.

While any participant who could read English could be suitable, two qualifying criteria were used to choose participants to rate the sentences. Firstly, that they were native English speakers. This removes issues of not understanding the meaning of the sentence that could arise with less fluent English speakers. Secondly, the participants were all chosen to have a science degree. This means that the participants would be familiar with the task of quantifying values on a scale.

For the ten pairs and thirty pairs datasets, two different groups of ten participants were used. This is few enough that a differing opinion could still be relevant to the similarity score but large enough to reduce some of the noise from where people judged the value to be within a range but had to make a single judgement on that occasion.

The introduction of a new similarity scale (section 3.7.1) for the opposites dataset means that potentially there could be situation where confusion could arise as to how the similarity of a pair of sentences were being rated. While it could arise that no opposite relationship that the similarity score would be the same as the original scale, this cannot be assumed to be certain.

In order to limit potential confusion with the scales while giving minimal instructions to avoid potentially prejudicing the sentences classed as opposite, the raters were restricted to people already familiar with sentence similarity tests and having done at least one on a prior occasion. This reduced the number of participants to five but the greater experience potentially lowered the noise on the experiments.

While there were not any comments from the raters suggesting confusion from the new scale, an earlier trial had encountered an inexperienced person wanting some clarification. This result was already excluded though due to an error in the responses meaning that the scores could no longer be aligned with the pair being rated.

While it was the mean value that was wanted, the tables later in this chapter also include the anonymous individual ratings from the participants, lest in the future a more refined

collation is desired.

3.6 New General Purpose Datasets

The first new datasets to be presented here can be described as general purpose because they aim to include variation in the core Linguistic principles that could apply to any sentence pairs, as opposed to specialist datasets where the sentence pairs can have a logical link such as the paraphrases or definitions used before. The datasets, however, are not regarded as comprehensive and many others of this ilk could be produced in the future to further expand the resources for testing sentence similarity models.

The datasets will exclude many special situations that could be relevant to similarity due to their sheer number and are being created to fill a gap identified in the test datasets that remain from the pre-existing datasets (sections 3.3.1 & 3.4.1). This was necessary for the evaluation of the sentence similarity models that will be created as part of this research.

The scale that is used is the standard 0 to 1 where 0 means that there is no similarity between the sentence pairs. The rating of the sentence pairs has already been discussed and the approach based upon the method given in section 3.5 will be expanded in this section.

3.6.1 Ten Pairs Dataset Construction

The 5 steps outlined in the method (section 3.5.2) can be used to describe the approach taken to create the 10 pairs dataset. The criteria (1) and (2) laid out in the general method are met through including Linguistic variation in the dataset.

The main requirement was that there is variation in the parts of speech used and in particular the main verb. These, however, were trivial constraints and would be expected to

be met as long as all the sentence pairs weren't the same special case which happened in the definitions used by the STASIS-30 dataset (O'Shea et al., 2008).

Other specific requirements were met through including sentences with connections that would instigate the needed variation. In terms of computational difficulty, the main concern comes from disambiguating meanings. The word similarity metrics underlying many of the sentence similarity models (see section 2.8) will select the highest overlap of meaning between two words. However, when words appear in a sentence they have a context that could be setting their meanings to different meanings than the most similar.

The inclusion of a connection between sentences in a pair, so that the same form of word could occur in both sentences but with different meaning, were included. As too were highly similar sentences where the words were still conveying the highest meaning overlap.

The remaining Linguistic variation wanted was that there was a variety of ways in which the words were combining, because of their grammatical function within the clauses and the sentence.

One exemplar as to the influence of punctuation which is well-known and can be found in Taggart and Wines (2008), was included. This was a sentence pair where the clauses were determined by the punctuation and significantly affecting the similarity of the meanings.

Other connections were used which could be very general such as descriptive imagery which had no specified inclusion of a Linguistic or computationally challenging feature.

All the sentence pairs include the core concepts being discussed from section 2.2 and the sentence similarity and meanings are built from multiple complex interactions happening at once. The inclusion criteria mean that the features would be more likely to make a distinguishable contribution to the overall similarity scores of the dataset.

The final step before the rating involved ensuring a probable range in the similarity scores. This was primarily done to reduce potential duplication resulting from having multiple sentences displaying the same characteristics. The ability to identify and use similarity is easier than the reverse. This means that conceptually, it is easier to conceive sentence pairs with either high or low similarity than mid level similarity. Low similarity can be achieved by having unconnected sentences and high similarity can be done by simply producing sentences with the same meanings.

The sentence pair creation had also included sentence pairs with likely mid-level similarity too. In order to maximise the variation of features contributing to the similarity, it is best to contain multiple examples of the similarity. This allows a model over or under estimating to be identified as well as where the influences of more multiple factors are combining.

To ensure a range the sentences were given a provisional rating and then using these selected 3 high (over 0.7), 4 medium (between 0.3 and 0.7) and low similarity (below 0.3). The range of similarity also made it easier for a person to think more widely about the possible types of similarity that could occur.

3.6.2 Ten Pairs Dataset

The resultant dataset along with the mean of the human scores is given in table 3.1 and the individual scores can be found in table 3.2. Sentence pair 9, a well-known example used as a demonstration of punctuation that can be found in Taggart and Wines (2008). This is a computationally complex pair of sentences that highlights the influence of clauses to the meanings. A more in-depth discussion of the dataset in relation to the model's performance is given in chapter 11.

ID	Sentence Pair	Mean of scores
1	<i>The Persian cat sat on the carpet. The ginger cat sat on the mat.</i>	0.78
2	<i>The caterpillar metamorphosed into an elegant butterfly. The caterpillar changed into a beautiful butterfly.</i>	0.91
3	<i>Fish swim in water. Birds fly in the air.</i>	0.27
4	<i>They believed the red bus was environmentally friendly. They put their faith in the train being green.</i>	0.44
5	<i>To drive a manual car, you must press down the clutch. To open the window, the mouse has to be double clicked.</i>	0.24
6	<i>The green grass glimmered as the sun shone on the morning dew. The ancient building had stood on that small hill for eons.</i>	0.07
7	<i>The Persian cat sat on the carpet. The Persian rug was on the dresser.</i>	0.18
8	<i>The exploded diagram shows how cars work. The car exploded at the art show.</i>	0.11
9	<i>Woman, without her man, is nothing. Woman: without her, man is nothing.</i>	0.37
10	<i>Trees need sunlight and water to grow. Food and drink are essential for your development.</i>	0.37

Table 3.1 Sentence pairs and mean of the human scores for the ten pairs dataset

ID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
1	0.6	0.5	0.5	0.9	0.9	0.8	0.9	0.9	0.9	0.9
2	0.9	0.9	0.9	1.0	0.7	0.8	1.0	1.0	0.9	1.0
3	0.3	0.1	0.5	0.0	0.8	0.3	0.0	0.1	0.6	0.0
4	0.8	0.2	0.4	0.2	0.6	0.6	0.3	0.3	0.4	0.6
5	0.2	0.1	0.4	0.1	0.8	0.1	0.0	0.1	0.6	0.0
6	0.2	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.2	0.1
7	0.1	0.1	0.2	0.0	0.3	0.3	0.0	0.0	0.6	0.2
8	0.1	0.0	0.2	0.1	0.1	0.1	0.1	0.1	0.2	0.1
9	0.0	0.0	0.8	0.2	0.8	0.1	0.4	1.0	0.3	0.1
10	0.4	0.1	0.7	0.3	0.8	0.3	0.1	0.5	0.5	0.0

Table 3.2 The individual scores for each of the sentence pairs in the ten pairs dataset from table 3.1 for the 10 participants labelled A1-A10

3.6.3 Thirty Pairs Dataset Construction

The thirty pairs dataset was simply an extension of the ten pairs dataset. It was created after the core experimentation had been completed to provide a larger long-term benchmark than for when a simple relative change from changing one aspect of a model was needed (as was enabled by the experimental approach given in chapter 5).

The extra sentences that were added, considered some of the features of complexity which had only limited variation in the 10 pairs dataset. The objective was to produce a similar level of complexity but avoid basic repetition, to ensure the concepts were being tested in slightly different respects to the ten pairs dataset. This provides extra validation of the earlier ideas that had been tested with the ten pairs dataset.

Just as the ten pairs dataset had included a pair from an external source two further pairs from the thirty pairs dataset also did the same.

A pair (pair 10) was taken from Mitchell and Lapata (2011) which they had used as an illustration of word order effecting the meaning. This again relates to variation in the word interaction when there is the same vocabulary. This example provides the dataset with a stronger signal for the contribution of the word interaction from the word order than would be present in an ordinary dataset but was already a feature that potentially contributes to all similarity scores.

The second set of sentences were not already aligned as a pair (pair 9) but were taken to give a relationship of descriptive language with closely related meanings but not talking about the same thing in the manner that was the case for the MSRP (Dolan et al., 2004). Both sentences were sourced from the easily available online newspaper, the Guardian, from closely related stories from the environment section (Guardian website, 2013) about declining insect numbers.

The remaining sentences were likewise selected as before to include a significant level of computational complexity and a range of similarities. The aim was to give a dataset with a consistent level of difficulty, as was the case with the ten pairs dataset but with larger variation.

Since the variation is still in the same fundamental concepts, that were already present in the ten pairs dataset, the ideas being tested remain the same. The difference between the two sets is mainly in the number of different ways that each concept occurs. This can potentially increase the signal from the features relative to the noise added from the human judgement.

3.6.4 Thirty Pairs Dataset

Table 3.3 gives the thirty pairs datasets and the human means. The individual human ratings are given in table 3.4. The ten pairs dataset was included as part of the thirty pairs and it can be seen that minor differences in rating resulted and this is discussed in section

3.6.5.

ID	Sentence Pair	Mean of scores
1	<i>The red car was illegally parked on the yellow line . The cake was eaten by the hungry boy.</i>	0.01
2	<i>The glass of water is on the table. The book was atop the dresser</i>	0.29
3	<i>I heard the birds singing in the morning. I like listening to birdsong.</i>	0.41
4	<i>The fast had lasted all day. The car was speeding for the whole journey.</i>	0.09
5	<i>The man who was standing by the river is the president of the company. My boss is standing beside the river.</i>	0.56
6	<i>The acrobats and tumblers were my favourite. I now need glasses to read my favourite book.</i>	0.16
7	<i>He shot the rifle at the rabbit. The woman photographed the giraffe.</i>	0.33
8	<i>The boat floats on the surface of the water. A hovercraft glides on a cushion of air.</i>	0.60
9	<i>Butterflies that flourish on grassland across Europe are in steep decline, indicating a catastrophic loss of flower rich meadows in many European countries. Wild populations of bumblebees appear to be in significant decline across Europe.</i>	0.48
10	<i>It was not the sales manager who hit the bottle that day, but the office worker with the serious drinking problem. That day the office manager, who was drinking, hit the problem sales worker with the bottle, but it was not serious.</i>	0.35

Table 3.3a Pairs 1-10 with human scores of the thirty pair dataset.

ID	Sentence Pair	Mean of scores
11	<i>The man hammered the brass hook into the wall. He screwed in the shiny screw.</i>	0.45
12	<i>The ice cracked beneath their feet The leaves rustled in the wind.</i>	0.14
13	<i>Water has been found on Mars! Water was found on the bathroom floor.</i>	0.60
14	<i>The rocket launched the satellite into orbit. The cement had held the bricks together for over a century.</i>	0.04
15	<i>Estate agents sell houses and flats. Greengrocers trade in fruit an vegetables.</i>	0.50
16	<i>The car was destroyed by a tree. The falling branch crumpled the automobile.</i>	0.75
17	<i>A passer-by was killed by a knife wielding maniac. The maniac stabbed a passer-by who died in hospital.</i>	0.85
18	<i>The barman had diluted the drinks. The owner of the pub had added water to the beer.</i>	0.81
19	<i>The sound of the violin brought tears to the audience. Music can sometimes make me cry.</i>	0.40
20	<i>The box was too small for the book to fit in. The men were fighting over the ticket.</i>	0.03

Table 3.3b Pairs 11-20 with human scores of the thirty pair dataset.

ID	Sentence Pair	Mean of scores
21	<i>The Persian cat sat on the carpet. The ginger cat sat on the mat.</i>	0.83
22	<i>The caterpillar metamorphosed into an elegant butterfly. The caterpillar changed into a beautiful butterfly.</i>	0.90
23	<i>Fish swim in water. Birds fly in the air.</i>	0.56
24	<i>They believed the red bus was environmentally friendly. They put their faith in the train being green.</i>	0.45
25	<i>To drive a manual car, you must press down the clutch. To open the window, the mouse has to be double clicked.</i>	0.30
26	<i>The green grass glimmered as the sun shone on the morning dew. The ancient building had stood on that small hill for eons.</i>	0.03
27	<i>The Persian cat sat on the carpet. The Persian rug was on the dresser.</i>	0.27
28	<i>The exploded diagram shows how cars work. The car exploded at the art show.</i>	0.07
29	<i>Woman, without her man, is nothing. Woman: without her, man is nothing.</i>	0.40
30	<i>Trees need sunlight and water to grow. Food and drink are essential for your development.</i>	0.39

Table 3.3c Pairs 21-30 with human scores of the thirty pair dataset.

ID	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
2	0.4	0.3	0.1	0.3	0.3	0.6	0.3	0.1	0.1	0.4
3	0.3	0.6	0.2	0.5	0.8	0.1	0.4	0.4	0.3	0.7
4	0.2	0.0	0.2	0.1	0.1	0.0	0.1	0.0	0.0	0.2
5	0.5	0.3	0.6	0.6	0.7	0.8	0.2	0.7	0.6	0.6
6	0.0	0.5	0.0	0.1	0.0	0.1	0.2	0.1	0.3	0.3
7	0.4	0.3	0.6	0.4	0.3	0.0	0.0	0.3	0.4	0.6
8	0.8	0.6	0.7	0.5	0.7	0.5	0.7	0.3	0.7	0.5
9	0.5	0.3	0.4	0.5	0.4	0.5	0.5	0.6	0.5	0.6
10	0.1	0.5	0.3	0.4	0.5	0.3	0.2	0.5	0.1	0.6
11	0.5	0.3	0.5	0.2	0.5	0.4	0.6	0.6	0.4	0.5
12	0.1	0.2	0.1	0.2	0.3	0.1	0.3	0.0	0.0	0.1
13	0.6	0.3	0.7	0.6	0.4	0.7	0.8	0.6	0.7	0.5
14	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
15	0.5	0.6	0.6	0.3	0.4	0.4	0.6	0.6	0.3	0.7
16	0.8	0.8	0.9	0.7	0.7	0.8	0.6	0.8	0.7	0.7
17	0.9	0.9	0.8	1.0	0.9	0.7	0.9	0.7	0.9	0.8
18	0.8	0.9	0.7	0.9	0.8	0.7	0.8	0.8	0.8	0.9
19	0.1	0.4	0.6	0.3	0.7	0.2	0.3	0.5	0.8	0.1
20	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0
21	0.8	0.7	0.9	0.9	0.9	0.9	0.7	0.8	0.9	0.8
22	0.9	0.8	1.0	0.9	1.0	0.8	0.9	0.9	0.9	0.9
23	0.6	0.6	0.7	0.7	0.4	0.5	0.5	0.6	0.7	0.3
24	0.6	0.4	0.2	0.6	0.8	0.2	0.4	0.3	0.3	0.7
25	0.2	0.5	0.2	0.3	0.6	0.4	0.3	0.2	0.2	0.1
26	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0
27	0.4	0.2	0.3	0.1	0.4	0.4	0.2	0.2	0.2	0.3
28	0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.1	0.0	0.2
29	0.4	0.0	0.7	0.1	0.8	0.4	0.6	0.1	0.7	0.2
30	0.4	0.6	0.5	0.7	0.5	0.3	0.4	0.5	0.2	0.3

Table 3.4 The individual scores for each of the sentence pairs for the thirty pairs dataset from table 3.3 for the 10 participants labelled B1-B10

3.6.5 Difference in Duplicated Sentence Scores

Differences between the ten pairs which were duplicated in both datasets were small and to be expected as human ratings are only an approximation of absolute similarity. Comparing the duplicated sentences' ratings (pairs 21-30) with the ratings for the ten pairs dataset it can be seen that most of the values are very close to each other. When judged as an overall set of data then the difference in the ratings would only have a minor impact on the correlation as the two ratings have a Pearson's correlation of over 0.94.

In principle, the ratings from the two different experiments could be combined together to form a single value as if 20 raters were shown the sentences. It would be reasonable to combine the scores in which case the two possible sets of answers offer an even closer correlation than with each other of about 0.985 on the Pearson's correlation.

These high correlations between the two sets means that experimentally the change in values between the human ratings has minimal impact upon the Pearson's correlations used as the primary method of judging whether an improvement was obtained. Chapter 11 examined how the performance of the model might have been affected by the Linguistic structures and the human ratings.

Although, it could be valid to merge the human scores, potentially the context of the sentence pairs in the dataset could have an impact on the human judgement of the semantic sentence similarity. The two groups were giving very close answers and without isolating whether the dataset did have an influence from its context, it is necessary from an experimental point of view, to treat the two sets of ratings as independent from one another.

The size of the variations between the two sets does suggest the level of repeatability to allow for both sets of values to be considered consistent with each other. It confirms that the method that has been adopted as standard in the literature is consistent, and that a level of repeatability on the scores would be expected in the future.

Essentially people were being asked to rate the similarity of the sentences in the context of all possible meanings when putting them on a scale of 0 to 1. It is not a realistic expectation that someone will be able to consider all possible meanings when making a judgement. As a result, the inclusion of a particular example set of sentence pairs could bring people's attention to possible meanings that they might otherwise have been overlooking. Conversely, were there not an increase in meanings, then there would be the possibility that a feature of the similarity could be overlooked from the dataset not containing a difference in these points and a person ignoring inherent common features between the sentences.

If you present the same pair of sentences to a person on different occasions it is not necessarily the case that they will return the same value each time. This can be seen with the Microsoft research paraphrase dataset (Dolan et al., 2004) when the identical pair appears twice and is classified once as a paraphrase and once as a non-paraphrase.

Therefore, it would perhaps be more accurate to view the given human score as lying within a range of possible values, rather than as a definitive answer that they would give every time of asking. The range that each person has for the possible scores that they might give is not necessarily the same. The result is that there can be some variation from which values the people with the same range would choose, on a particular occasion, and some additional variation as a result of people either having a tighter range or a different central value.

This means that there is essentially noise from randomness, rather than any deterministic factor that could account for the differences in scores between any two sets of raters.

It is likely though, that some of the variation between the two sets was the result of a combination of the participants' initial ability to rate similarity and having been given more sentences to judge. While both groups of raters fulfilled the loose criteria of being a native English speaker and a holding degree in science (so as to give experience at

quantifying things on scales), the second group (for the thirty pairs dataset) potentially contained individuals who would be expected to perform better than a random sample of people who met the inclusion criteria, as this included people who were experts in board games. In addition they were seeing more sentence pairs as part of the dataset that they were rating, than the ten pairs participants.

While there are obvious cognitive science questions about why a particular person chose to rate a particular pair of sentences with a particular value, there is minimal impact that this understanding has, on whether the scores are indicative or related to the absolute similarity (section 1.6.1). It is likely that there was only a marginal improvement from the second set of raters and that most of the differences were just down to chance.

In terms of the experiments and focus of this research it is not necessary to have a deeper understanding of why the human ratings were what they were, only that they can be used as an approximation of the absolute similarity. Since, the purpose of the ratings is this approximation of absolute similarity, it does not matter whether a group is giving better estimation than would be expected from the inclusion criteria alone as it would simply mean that there was less noise on the values.

Taking the mean reduces the variance from the any randomness of choice within a person's range of values. It can lead to situations when there are distinct grouping of non-overlapping ranges that the distinctiveness of the answers is reduced.

This is an observation that can be made for sentence pair 29 (pair 9 in the ten pairs dataset). Although both sets of scores considered as a group have very close means to each other, there was a large range of values being given as scores. This pair is discussed in the analysis in Chapter 11, but as a result of there being an level of oppositeness in the meaning the rating on the standard sentence similarity scale can be confused.

One other pair also discussed in chapter 11 is pair 23 (pair 3 in the ten pairs dataset). It can be seen that this one pair shows a significant difference between the two sets of mean

values. This is not simply attributable to random chance and there is little ambiguity in the interpretation of the meaning of the sentences. This suggests that the second group is judging the value significantly higher than the first group. In chapter 11, this was suggested that the ten pairs rating for this sentence pair could have been lower than anticipated. The combined value from the 20 raters would be much less of a difference.

The higher similarity from the second group are finding a connection in the similarity from the general meaning that is not present superficially, if just looking for links between the words individually.

Sentence pair 27 (pair 7 in the ten pairs dataset) is also perhaps showing a difference between the two groups' ratings. In this case the sentences again have little potential ambiguity in their meanings, but in this case this was not highlighted in the analysis to come in chapter 11.

It is perhaps the case as the pair are examined with differing degree of depth that the judgement of similarity is altered. Superficially, the structure of the sentences are the same and there are a number of overlapping words. Other meanings between words are close which would predispose people towards giving a higher score.

When the meanings of the pair of sentences as a whole are examined, it can be seen that most of the features that were leading to the superficial similarity, are having far lower effect on the similarity of the pair, than the differences leading to a much lower similarity rating.

However, while the initial inclination would be to give a low rating (as can be seen from several 0.0 in table 3.2), a closer examination shows that there is still much overlap remaining between the two meanings. Both discuss an object located upon the upper surface of an item of furniture. This suggests that it is the slightly higher similarity which is more representative of the meaning than was the case in the first example.

While clear differences arise in terms of the repeatability of the human ratings and how much noise is likely present, essentially the differences are not going to substantially alter the performance of a sentence similarity model on the thirty pairs dataset. It was known that the human scores were only approximating the actual desired answer. The differences suggest that perhaps a more refined combination than simply taking the mean might be a consideration for the future. Also it might be the case that more experienced raters would give a less noisy set of ratings.

3.7 Opposites

The opposite relationship (section 2.7.4) is one that can be very significant to similarity. While opposites can represent meanings that are as far apart in a particular respect as possible, they also can share significant overlap of meaning to allow them to be on the same scale.

As part of the experimental development in this thesis using a Linguistic approach, oppositeness is introduced to a sentence similarity model in chapter 14. This was an entirely new and significant step for the field of sentence similarity and required the creation of a new similarity scale, compared to the standard scale being used for the other new datasets.

This section presents the new scale, but it will be discussed further in Chapter 14, where the context of how its inception is a natural progression from the Linguistic approach will be clearer. Then as was the case for the other new datasets, the approach will be briefly discussed with respect to the 5 steps outlined in section 3.5.2. Finally, the new dataset itself and the individual raters scores is presented.

3.7.1 New Scale for Opposites

The opposite relationship was outlined in section 2.7.4 as for how meanings could be opposites. An opposite relationship means that two meanings could be placed on a scale and that the meanings were on the extreme end of the scale.

The original scale uses 0 for when the sentences were unconnected, yet, for opposites there is also a significant overlap from the meanings being on the same scale. From a similarity perspective the meanings of opposites are even further apart than if there had been no connection between the meanings, or overlap of meaning.

There is a simple intuitive method to handle opposites while still retaining a score of 0 when there is no similarity between the sentences. This is to introduce a sign to the similarity. Negative numbers can indicate that the meanings were opposite to some extent and the magnitude or strength of the similarity.

As opposed to the standard similarity measure rated between 0 and 1 for similarity, the output is ranged from -1 to 1 and in essence a separate scale. It remains the case that the lowest similarity is 0 but that the meanings can be further apart despite having overlap to their meaning.

While this step seems natural and the scale logical, it is never the less a new step for sentence similarity. The scale is discussed again in chapter 14 when opposites are introduced into a sentence. The specialist dataset for opposites, simply named the opposites dataset, is scored using this new scale.

3.7.2 Constructing the Opposites Dataset

The final dataset being created was using the new similarity scores for the rating and was rated by people whom already possessed prior experience from participating in sentence similarity tests (section 5.5.1). The opposites dataset is a specialist dataset and while there is still a desire to contain a level of variation of the Linguistic concepts as had been the

case for the ten pairs and thirty pairs datasets, the main interest was in distinguishing the impact of opposites.

There were specific criteria to include at least one example of each kind of opposite and to include several examples of pairs that could be near opposite. This meant that alongside the opposite relationships (section 2.7.4) of inversion, antonymy and negatives, that sentence pairs with contradictions and containing ideas that could be expressed on a scale were included. In addition, some sentence pairs where the word interaction included logical differences, such as conditionals or questions, were also included.

The reason for this was to provide both the computer model with challenging test cases and to provide several examples to the human raters where they might potentially feel that the pairs would rate differently on the different scale.

In conjunction with these sentences, pairs were included where there were variation in how the ideas combined together and issues such as ditransitive verb clauses which could challenge the task of identifying inversion.

The dataset still contains significant Linguistic variation but perhaps slightly easier to process than had been the case from the earlier examples (within the thirty pairs dataset) in which the Linguistic exemplars had a less direct relationship to similarity, as arises from the opposite relationships. The comparison of similarity is still non-trivial as too would be the separate task of identifying the opposites from within the dataset.

3.7.3 Opposites Dataset

Table 3.5 includes the individual scores for each sentence pair using the new similarity scale and the mean of the scores to be used as the approximation of the absolute similarity.

Pair	Sentence						Mean
1a	<i>Matthew gave the charity money.</i>	0.9	0.85	0.9	1	0.8	0.89
1b	<i>Matthew donated to the charity.</i>						
2a	<i>I love ice-cream.</i>	-0.95	-0.95	-1	-0.8	-0.8	-0.9
2b	<i>I do not love ice-cream.</i>						
3a	<i>I cannot paint horses.</i>	0.35	0.25	0.25	0.3	0.3	0.29
3b	<i>I drew the picture.</i>						
4a	<i>I do not run unless I have too.</i>	-0.1	-0.35	0.2	-0.4	0.3	-0.07
4b	<i>I run all the time.</i>						
5a	<i>I won the race.</i>	0.75	0.8	0.65	0.75	0.6	0.71
5b	<i>I will win the race.</i>						
6a	<i>The favourite should win the race.</i>	0.6	0.45	0.45	0.55	0.4	0.49
6b	<i>A newcomer could win the race if the favourite falls.</i>						
7a	<i>The favourite should win the race.</i>	0.6	0.45	0.45	0.55	0.4	0.49
7b	<i>Should the favourite fall a newcomer could win the race.</i>						
8a	<i>You can do it.</i>	0.7	0.7	0.6	0.9	0.7	0.72
8b	<i>You did it.</i>						

Table 3.5a: Pairs 1- 8 of the opposites datasets with the 5 individual scores and mean of the scores

Pair	Sentence						Mean
9a	<i>The big man would make a good king.</i>	0.3	0.1	0.45	0.2	0.5	0.31
9b	<i>For the sake of the country, Mark must rule.</i>						
10a	<i>I must finish.</i>	0.35	0.6	0.65	0.55	0.6	0.55
10b	<i>I might finish.</i>						
11a	<i>Cats enjoy sleeping.</i>	0.95	1	0.95	1	1	0.98
11b	<i>A cat enjoys sleeping.</i>						
12a	<i>The oak tree will grow in the spring.</i>	0	0.1	0.2	0	0.1	0.08
12b	<i>An asteroid will hit soon.</i>						
13a	<i>Do the dishes.</i>	1	1	0.95	0.85	0.8	0.92
13b	<i>Wash-up.</i>						
14a	<i>Hear the birds singing.</i>	0.05	0.25	0.35	0	0.3	0.19
14b	<i>Smell the scent of the roses.</i>						
15a	<i>The dog fighting in the street is black.</i>	0.8	0.3	0.3	0.7	0.7	0.56
15b	<i>The dog fights in the street.</i>						
16a	<i>The man is too old.</i>	0.55	0.35	0.8	0.6	0.65	0.59
16b	<i>The woman declared, "the man is too old."</i>						

Table 3.5b: Pairs 9- 16 of the opposites datasets with the 5 individual scores and mean score

Pair	Sentence						Mean
17a	<i>Elephant</i>	0.65	0.5	0.7	0.45	0.3	0.52
17b	<i>Mouse</i>						
18a	<i>The big grey animal</i>	0.9	0.85	0.85	0.85	0.8	0.85
18b	<i>An elephant</i>						
19a	<i>The black cat is drinking from a saucer of milk.</i>	0.55	0.7	0.8	0.85	0.8	0.74
19b	<i>The white cat is drinking milk from a saucer.</i>						
20a	<i>The small elephant.</i>	0.9	0.8	0.8	0.8	0.8	0.82
20b	<i>The Elephant.</i>						
21a	<i>The old teacher gave the boy the money,</i>	0.75	0.75	0.55	0.6	0.6	0.65
21b	<i>The professor gave treasure to the girl.</i>						
22a	<i>You, the Roman people, are an inspiration.</i>	0.85	0.8	0.75	0.75	0.75	0.78
22b	<i>The Romans are an inspiration.</i>						
23a	<i>The woman ran.</i>	0.75	0.75	0.7	0.65	0.7	0.71
23b	<i>She goes quickly.</i>						
24a	<i>The warrior whipped the slave.</i>	0.75	1	0.95	0.7	0.9	0.86
24b	<i>The warrior used the whip on the slave.</i>						

Table 3.5c: Pairs 17- 24 of the opposites datasets with the 5 individual scores and mean score

Pair	Sentence						Mean
25a	<i>The slave was whipped by his master.</i>	-0.9	-0.85	-0.8	-0.85	-0.8	-0.84
25b	<i>The slave was not whipped.</i>						
26a	<i>The hero won the fight.</i>	-0.85	-0.8	-0.95	-0.85	-0.8	-0.85
26b	<i>The villain won the fight.</i>						
27a	<i>The hero won the fight.</i>	-1	-1	-0.95	-0.9	-0.75	-0.92
27b	<i>The hero lost the fight.</i>						
28a	<i>The hero slew the villain.</i>	-0.8	-0.9	-0.85	-1	-0.8	-0.87
28b	<i>The villain slew the hero.</i>						
29a	<i>Cats do not like to be stroked backwards.</i>	0.25	0	0.1	0.1	0.2	0.13
29b	<i>The man yelled with pain when he hit his thumb.</i>						
30a	<i>The cameraman shot the wedding.</i>	0.35	0.25	0.2	0.25	0.4	0.29
30b	<i>The sniper shot his enemy.</i>						
31a	<i>The dog ate the bone.</i>	0.95	1	0.95	1	1	0.98
31b	<i>The bone was eaten by the dog.</i>						
32a	<i>It was raining.</i>	0.1	0.25	0.3	0.15	0.4	0.24
32b	<i>I hate daylight.</i>						

Table 3.5d: Pairs 25- 32 of the opposites datasets with the 5 individual scores and mean human score

Pair	Sentence 1						Mean
33a	<i>The cat drank lemonade.</i>	0.1	0	0.1	0	0.55	0.15
33b	<i>My good friend only eats bananas.</i>						
34a	<i>The ice-cream children like the best is vanilla.</i>	0.35	0.6	0.65	0.7	0.6	0.58
34b	<i>The children wanted ice-cream.</i>						
35a	<i>What is the answer?</i>	0.3	0.55	0.4	0.45	0.6	0.46
35b	<i>The answer is 6.</i>						
36a	<i>Won't you come to the game?</i>	1	1	1	1	1	1
36b	<i>Will you come to the game?</i>						
37a	<i>Don't you dare do it!</i>	0.8	0.75	0.9	0.65	0.85	0.79
37b	<i>Don't you dare do it?</i>						
38a	<i>The butter had been left out too long.</i>	1	1	1	0.95	1	0.99
38b	<i>The butter has been left out too long.</i>						
39a	<i>The first time that I saw her I knew I was going to marry her.</i>	0.6	0.6	0.65	0.7	0.65	0.64
39b	<i>I am engaged to my fiancée.</i>						
40a	<i>The water makes them thirstier.</i>	0.3	0.35	0.25	0.25	0.3	0.29
40b	<i>The dry desert is very hot.</i>						

Table 3.5e: Pairs 33- 40 of the opposites datasets with the 5 individual scores and mean score

Pair	Sentence 1						Mean
41a	He imagined winning the race.	0.75	0.6	0.85	0.65	0.6	0.69
41b	He wins the race.						
42a	He should win the race.	0.55	0.65	0.55	0.55	0.65	0.59
42b	He imagined winning the race.						
43a	The slow car went to London.	0.9	0.8	0.9	0.8	0.8	0.84
43b	The man drove slowly to London.						
44a	The man shot the rabbit.	0.75	0.65	0.6	0.7	0.65	0.67
44b	The hare was injured by the boy.						
45a	He whipped the slaves.	0.7	0.75	0.8	0.85	0.85	0.78
45b	He used the whip on the horse.						
46a	My donation was welcomed by James.	0.75	0.85	0.65	0.55	0.75	0.71
46b	I gave James the money.						

Table 3.5f: Pairs 41- 46 of the opposites datasets with the 5 individual scores and mean human score

3.8 Conclusions

This chapter has included what is potentially very significant for the field of sentence similarity; the new similarity scale for use with opposites. This is an area that is examined in far more detail in chapter 14. The scale is intuitive and simple but the idea of treating opposites as a special case had not been encountered before. The idea became a natural one as part of the Linguistic approach adopted by this research, which will become more apparent later. It is included here so as to present all of the new datasets in one place.

The chapter started identifying the standard datasets and discussing their limitations with respect to the experiments needed to test the objective of this research from section 1.4.

The other parts of the chapter were detailing the datasets that would be needed for later in the research, in order to test the viability of sentence similarity. The three new datasets were mainly following the methodology already used but with extra refinement to provide the Linguistic variation, which was unfortunately missing from the definitions and paraphrase datasets. The datasets, while enhancing the available future resources for evaluating sentence similarity, are largely only peripheral to the field of sentence similarity itself.

The next chapter presents the framework that is the foundation of the sentence similarity model to be developed to include Linguistic components.

4.0 Linguistic Framework

4.1 Introduction

The objective of this research is to show that the inclusion of Linguistic concepts to a sentence similarity model could lead to an improvement in accuracy. Before this can be pursued, it is necessary to build a model that is capable of incorporating an implementation of the Linguistic concepts being investigated. Additionally, it is desirable to be able to introduce the components gradually, so that the effect from each different concept being added can be considered individually.

To this end, an extensible modular framework was conceived so as to define the purpose of each module, with respect to the Linguistic concepts needed for sentence similarity. The creation of the framework uses the Linguistic concepts adapted for sentence similarity described in section 2.2. This is a fundamental step for sentence similarity as it adopts a Linguistic approach while enabling the potential computational automation of the tasks. The Linguistic approach being taken is novel for sentence similarity and therefore the framework and its extensibility are also novel.

The modules correspond closely to Linguistic concepts but in some cases can only approximate the human approach to language. Since sentence similarity is intrinsically an approximate approach, this was expected. The framework includes the capability to include Linguistic features beyond the scope of this research. For example, there is a grouper module to handle compound sentences and a combiner module which allows for the handling of special Linguistics (cases such as conditional sentences). However, the model development will only use the general concepts which can be applied to any sentence pair until the final development in chapter 14.

It will be seen in later chapters as the experiments progressed, that the framework enabled easy extension of the mathematical model created as the original implementation of a sentence similarity model adhering to the framework (presented in chapter 6).

This chapter presents the framework and modules in the general case, where there are several possible valid approaches to implementing the modules that meet the requirements. However, in the actual implementation, a narrowing of options is adopted to reflect the design choices. No specific sentence similarity model is created from the framework in this chapter. It is only the framework itself presented with the potential for a model to be created.

4.2 Framework

Having detailed the experimental methodology in chapter 4, the next step is to provide the framework which will form the backbone of the sentence similarity model and allow for the inclusion of core Linguistic elements.

The framework presented here, is constructed to include the key Linguistic principles rather than simply for the specific purposes of the sentence similarity model needed for this investigation. It is a general framework. The source of knowledge for the meanings is first used to find the word similarities before being combined to find the key concepts as detailed in chapter 2. When the sentence similarity model is referred to later in this chapter, it refers to a general instance of a sentence similarity model that is adhering to the framework.

Prior to sentences being input to a sentence similarity model adhering to the framework, it is possible that either the sentence is tagged with additional information added to each word (such as to give a specific meaning or part of speech) or that the knowledge base has been updated with temporary vocabulary. Otherwise, it is assumed that only context of comparison applies and that the same interpretation of the intended meaning should occur every time a pair of sentences are input and with the same absolute similarity score.

Therefore, by definition, the model is commutative and the maximum sentence similarity score should occur when a sentence is compared against itself. In the most general terms the semantic sentence similarity model works like a black box so that:

$$f(\text{sentence 1}, \text{sentence 2}) = \text{Semantic Similarity Score} [\text{min} : \text{max}]$$

[4.1]

The output of the model and word similarity module (defined later) is on the same range between the minimum and maximum value. While this will be the same for a particular implementation, this can be using differing scales and still have the same architecture.

A modular framework is used with modules themselves closely relating to Linguistic concepts. The key concepts of context, word interaction and topic are used in conjunction with the word similarities in order to convert the underlying knowledge source into a single similarity score. The knowledge source is the information which can be used to decide how to compare meanings or ideas that the model is already aware of. For example, a method of how judge the similarity of a “car” to a “lorry”.

Figure 4.1 shows how two sentences' information passes through the key modules. Knowledge is added via two databases: a knowledge-base and a corpus. Individual word meanings are given a similarity from the Word Meaning Similarity module to each possible word meaning in the other sentence before being combined with weights. This information is finally converted by a block of modules in the Algorithm to give a single similarity score.

The framework comprises 9 core modules:

- Grouper
- Knowledge Base – (see section 4.3)
- Corpus – (see section 4.4)
- Weighting – (see section 4.4)
- Word Meaning Similarity – (see section 4.3)
- Parser – (see section 4.4)
- Topic Similarity – (see section 4.5)
- Word Interaction Similarity – (see section 4.5)
- Combiner – (see section 4.5)

The Knowledge base and Word Meaning Similarity (WMS) module combine together to give the similarity scores for all the possible meanings. By selecting the highest possible meanings for a pair of words, this would function as a word similarity model.

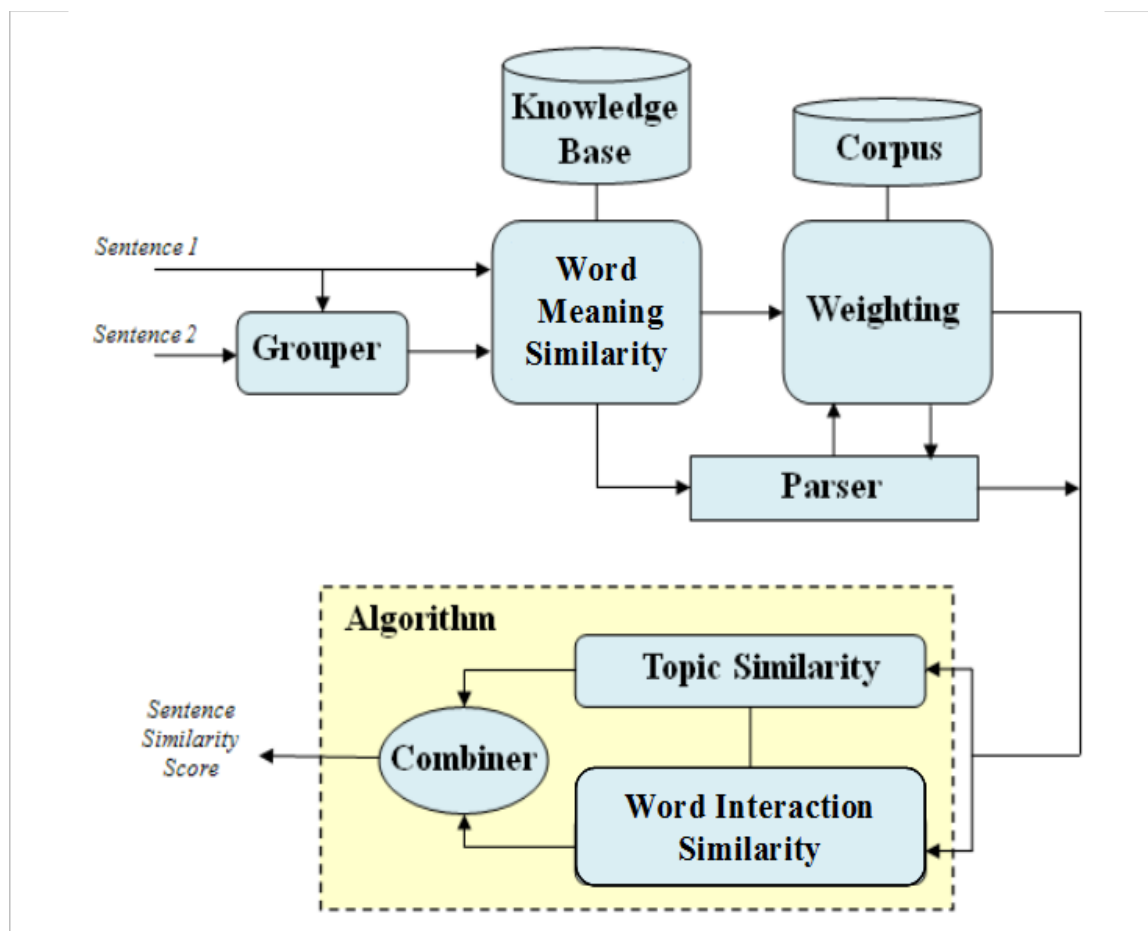


Figure 4.1 Linguistic Sentence Similarity Framework

The Weighting Module uses the Parser and Corpus to add the context: to distinguish the intended meaning and significance to the similarity that each meaning combination could be having on the overall similarity.

The topic similarity, word interaction similarity and combiner modules function together to form the algorithm module. This takes the individual scores for all the possible word combinations from the WMS, the context and structural information added together to form the single similarity score for the sentence. This involves examining how the meanings of the words function to give the overall meaning.

The Grouper module is included purely for completeness, but is not used for any of the versions of the model discussed in this thesis. Where two sentences have been merged with a conjunction into a single sentence, the sentence can often be rewritten without altering its meaning. Such sentences can be rewritten by the grouper so as to ensure a like for like comparison.

"The cat ran and jumped."

"The cat pounced and the cat skidded."

Here, to ensure a like for like comparison, the second sentence is altered so that it has the same format as first sentence. In this case the duplicated subjects are merged to give a grouped version of the second sentence as:

"The cat pounced and skidded."

In practical scenarios, it is more likely that the function of a grouper would be included in a pre-processing module that might already choose to split a sentence with a simple conjunction like “and” in to two sentences.

The sentence similarity framework can be thought of in three stages:

- 1) Word Meanings - identify the meanings of words and find the similarities between all possible pairs of words between the two sentences.
- 2) Context - add weights and tags to each word so as to represent its contribution and function to the overall similarity.
- 3) Algorithm – convert a matrix of scores (formed from each combination of word meanings), tags and weights to obtain a single similarity score.

For the purpose of the next few sections a running example using the following inputs will be used:

Sentence 1 = *The ginger cats drank the milk.*
Sentence 2 = *My homework was eaten by my dog.*

4.3 Word Meanings

The knowledge base needs to take every word and return the known possible meanings and part of speech in a structure, for each word meaning. A structure can be any representation that can be compared for its similarity, as mentioned in section 2.6.1. An example of which would be a chain of hypernyms. The structures can then be used by the word meaning similarity module to give a similarity measure for a pair of meanings.

The knowledge base can be further split into a fixed lexical database and a temporary database. The temporary database can include any meanings (or exclusions) that are only relevant for the current context.

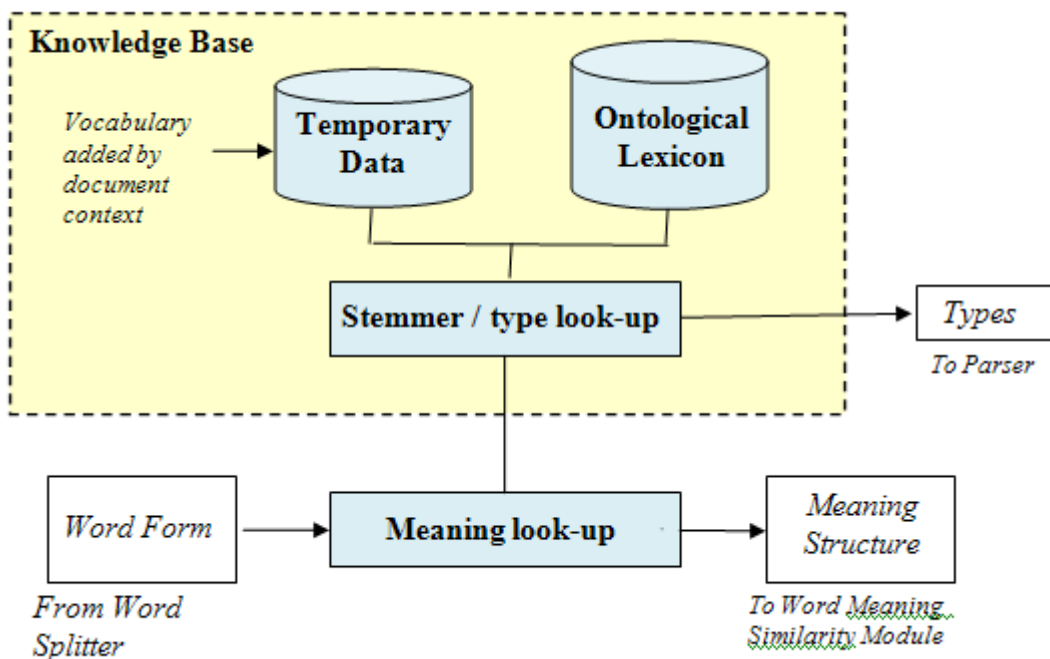


Figure 4.2: Knowledge Base Modules

Figure 4.2 shows the expanded knowledge base modules. It is showing the data flow from the input word to the output of the meaning structure and possible parts of speech via a

stemmer or look-up, depending on the format of the ontological lexicon. The knowledge-base is stored in two databases. One database can handle temporary meanings added for the current context such as a document and the other handles the fixed vocabulary.

For example, consider a situation where the word “cats” is being shown to the knowledge base. The meaning look-up needs to find all of the possible meanings that are in the lexical database. In this case there are three possible meanings of “a whip”, “a feline” and the obscure verb “to vomit”. Each of these meanings would use one of the meaning structures described in section 2.7.1. So for the sake of this example assume a hypernym chain where a “-” is used to represent a hypernym link. The final step is the stemmer / type look-up which needs to identify the verb and noun as possible parts of speech.

Word meaning look-up for the word “cats” may start by finding the stem and meaning structures:

stem = cat

valid type = noun plural or verb 3rd singular

cat#noun#1 = "cat"-*"feline"*-*"mammal"*-*"animal"*-*"life form"*-*"thing"*

cat#noun#2 = "cat"-*"whip"*-*"weapon"*-*"tool"*-*"thing"*

cat#verb#3 = "cat"-*"vomit"*-*"purge"*-*"divide"*-*"change"*-*"do"*

Similarly, the other words would be defined and their values and possible parts of speech passed on to the other modules.

So "dog" could give the following structures.

dog#noun#1 = "dog"-*"hot dog"*-*"food"*-*"fuel"*-*"thing"*

dog#noun#2 = "dog"-*"canine"*-*"mammal"*-*"animal"*-*"life form"*-*"thing"*

dog#verb#3 = "dog"-*"follow"*-*"movement"*-*"do"*

Next the meaning structures for cat would be used to compare “cats” to the possible meanings for the words in the other sentence.

The word meaning similarity module takes a pair of meaning structures (section 2.7.1) and gives a similarity score on the same scale as the overall sentence similarity model.

The meaning structures (enumerated 1 & 2) can be first converted into the overlapping meaning (common meaning) C12 and their distinct meanings D1 and D2.

$$f(D1, C12, D2) = \text{Meaning structure similarity [min : max]} \quad [4.2]$$

For the example assume that the meanings of dog#2 and cat#1 are being compared then:

$$\begin{aligned} \text{dog\#2} &= \text{dog} - \text{canine} - \text{animal} - \text{life form} - \text{thing} \\ \text{cat\#1} &= \text{cat} - \text{feline} - \text{animal} - \text{life form} - \text{thing} \end{aligned}$$

The the common meaning C12 is: *animal- mammal - life form - thing*

D1 is: “dog” - “canine” and D2 is “cat” - “feline”

The similarity score for this pair of meanings might well be given as 0.67 if using a simple ratio between overlapping terms, although a range of formula for [4.2] could be used.

This process is repeated to compare all of the possible meanings for sentence 1 against all the possible meanings in sentence 2. The output of the first stage is a matrix of similarity scores. Each score represents the comparison of a pair of meanings. The row and column for each meaning in the comparison can be used to give the type (equivalent to the part of speech) and the form of the word (its written representation from which the meaning was obtained). The form and type of each meaning in the comparison is also included as information passed through to the next stage of the algorithm.

Figure 4.3 shows how the matrix of similarity scores would be constructed for each of the possible meanings. Each meaning such as E4 also can have a type tag added and each word such as Word B, has a form associated with it.

Alongside forming a matrix, the possible types / parts of speech for each word are sent to

the parser module which is part of the context module detailed in section 4.4. The parser adds further tags to indicate the grammatical subdivisions of each input sentence. For the example pair being used this could include information along these lines:

Sentence1: Subject clause {the ginger cats} [article, adjective, noun]

Verb clause {drank} [verb past tense]

Object clause {the milk} [article noun]

Sentence 2: Object clause {my homework} [possessive pronoun, noun]

Verb clause {was eaten by} [aux. past participle, preposition]

Subject clause {my dog} [possessive pronoun, noun]

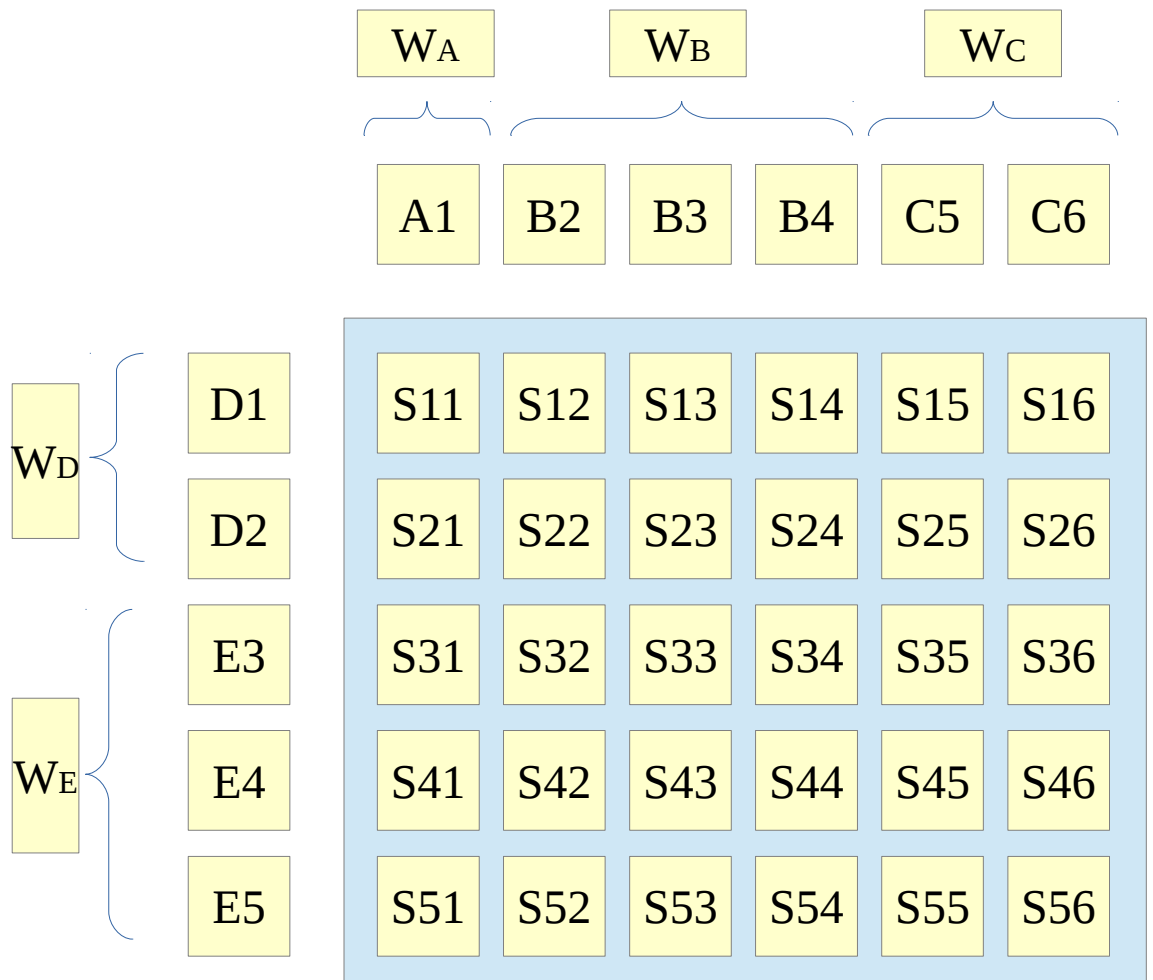


Figure 4.3: An illustration of the weights for sentence1 (Word A, Word B, Word C) and sentence 2 (Word D, Word E).

Importance weights, (shown in []) based upon its part of speech (shown as a letter the #)

signify the contribution of each possible meaning to the sentence, are given for a subsection of the matrix of similarities for the Subject clauses, then as an example, values similar to those shown below might result:

		The#t#1	Ginger#j#1	Ginger#n#2	Cats#n#1	Cats#n#2	Cats#v#3
		[0.3]	[0.6]	[1.0]	[1.0]	[1.0]	[1.4]
My#pos#1	[0.3]	0.30	0.10	0.00	0.00	0.00	0.00
My#int#2	[0.4]	0.00	0.00	0.00	0.00	0.00	0.00
Dog#n#1	[1.0]	0.00	0.00	0.60	0.18	0.25	0.00
Dog#n#2	[1.0]	0.00	0.00	0.18	0.67	0.18	0.00
Dog#v#2	[1.4]	0.00	0.00	0.00	0.00	0.00	0.20

4.4 Context

As mentioned in section 2.3.3, context can be very challenging to find without an understanding of the underlying meaning. Therefore, instead of directly including a context module, weights are added to represent the chosen meaning and its contribution to the overall meaning. Just as if a person were reading the sentence, each word is attributed a specific meaning.

First a decision is made as to meanings that are not possible in the given context. This information can be an example of the type of the word determined by the parser or even from human tagged inputs. Where a meaning is judged to be incorrect for the context, it will be given a weight of 0. This effectively, removes the meaning from being considered with respect to the overall similarity.

If the disambiguation weights were considered for Word B in figure 4.5 the meaning B2 were a noun and the parsing information meant that only verbs were possible, then weight w_4 would become 0. When the disambiguation weights are combined with the similarity scores, the meaning contributions from B2 to A1 and from B2 to A2 become 0.

It is possible to use a combination of two or more meanings to represent the context of a word in a sentence and then a distribution of the weight can be used. However, it is assumed that a single meaning is wanted for each word for the purposes of this thesis.

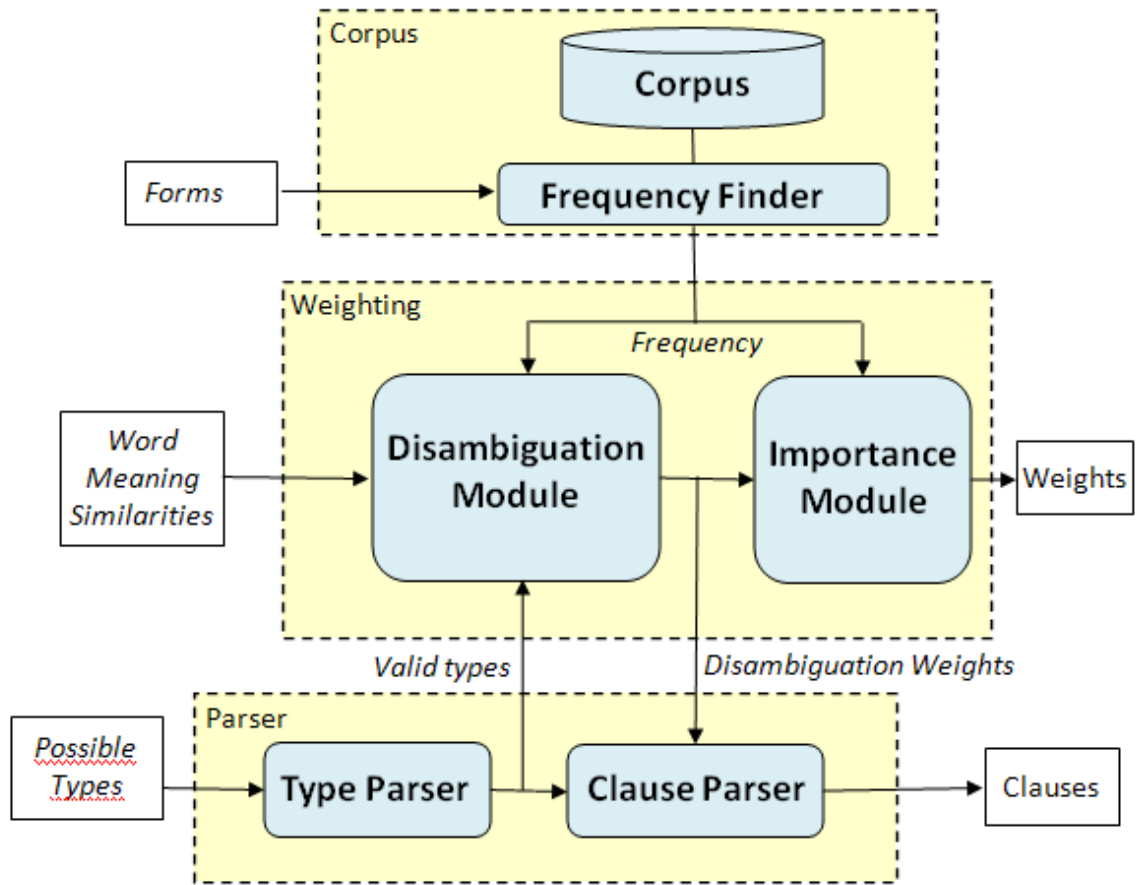


Figure 4.4: Context and Weighting Modules.

Corpus is included (for example, Brown's Corpus (Francis and Kucera, 1979) and could be used to help determine the most likely meaning from its usage, based upon the frequency of occurrence in the corpus.

The next stage is determining which meaning to select for each pair. This can be done using automatic disambiguation (using the knowledge from the corpus or a definition) or through the use of human selection of intended meaning.

When the disambiguation still results in more than one possible meaning, then the disambiguation is resolved using the possible values within the matrix of word meaning similarity scores for the sentence pair. The highest scoring meaning is selected as the

intended meaning. This is consistent with the context of comparison, where a person will assume the most similar meanings and a common context for the words.

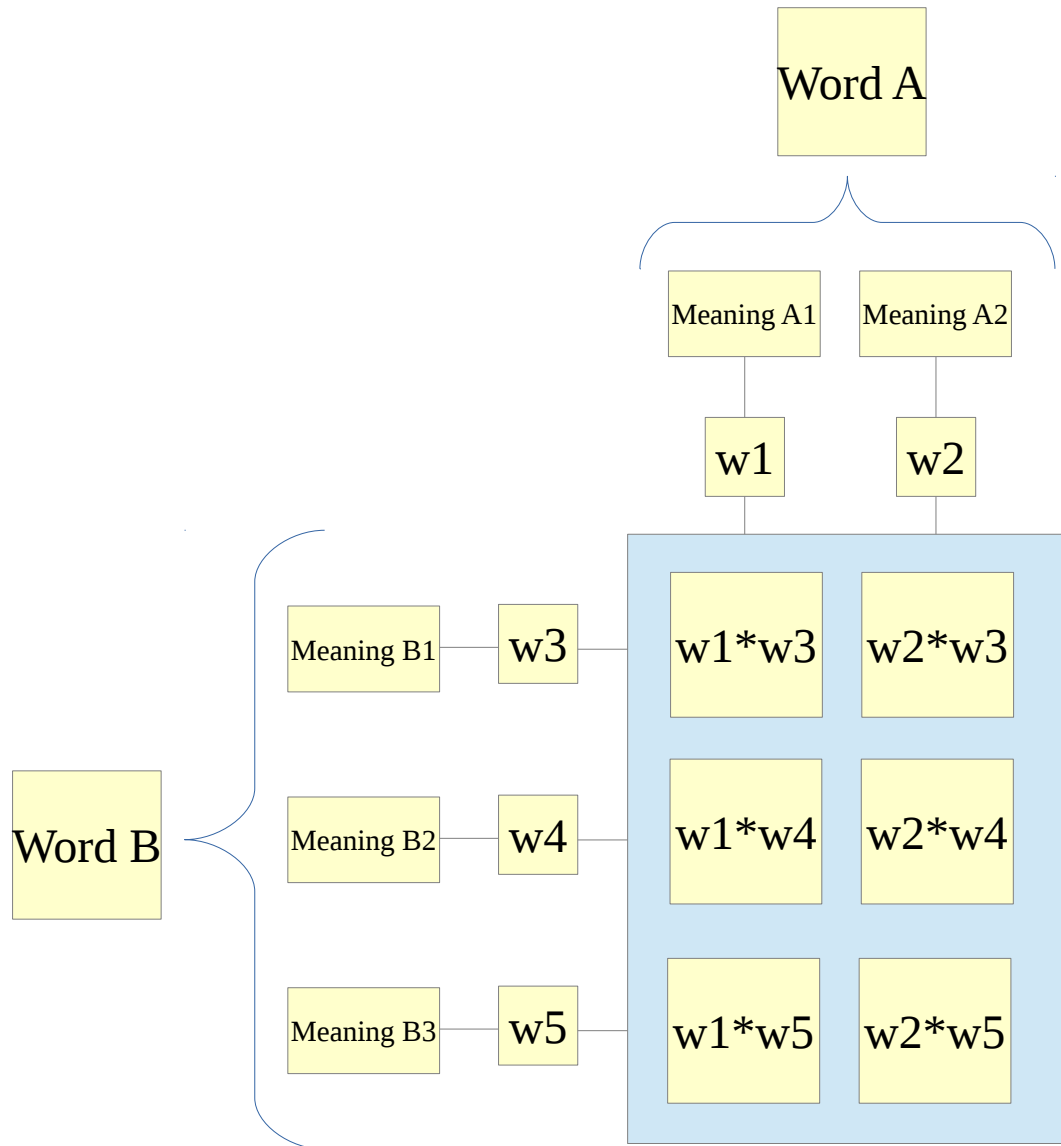


Figure 4.5: Diagram showing how the weights would be combined for two words.

A second weight is added to each meaning, the relative contribution to the overall meaning of the pair of a sentences. The result is a meaning that is making a larger contribution to the total meaning, will take a larger value for its importance weight. This means that the similarity from this meaning will have a bigger contribution to the overall similarity score. This assigns a weight based upon the word type and relative semantic content of each word which is to be estimated using the corpus information.

This in effect, adds two weights to every word. The final component of the context is the

addition of the clause information for each word, which adds a tag to give the type and clause type for every word. When a weight is applied to a word or a clause it could equally well be described as an identical weight for each of the possible meanings.

Returning to the example pair of sentences and the two subject clauses (“the ginger cats” & “my dog”), there are multiple possible meanings for all the words other than “the”. The word “ginger” is being considered with a noun meaning (as in the spice) and as an adjective (meaning the colour). Without disambiguation `Ginger#n#2` would pair with `Dog#n#1` as this is the highest similarity for the word ginger. Looking back to section 4.3, it can be seen that the parsing only allows an adjective so each of the meanings could be given a disambiguation weight:

<code>The#t#1</code>	<code>Ginger#a#1</code>	<code>Ginger#n#2</code>	<code>Cats#n#1</code>	<code>Cats#n#2</code>	<code>Cats#v#3</code>
[1.0]	[1.0]	[0.0]	[1.0]	[1.0]	[0.0]

<code>My#pos#1</code>	<code>My#int#2</code>	<code>Dog#n#1</code>	<code>Dog#n#2</code>	<code>Dog#v#3</code>
[1.0]	[0.0]	[1.0]	[1.0]	[0.0]

Further disambiguation could potentially be added for “cats” as a feline, via a possible association with the word “milk.”. The disambiguation weight changes for all the meanings for cats other than `Cats#n#1` to 0.0., leaving just “dog” to be resolved by nearest similarity.

4.5 Algorithm Module

The final stage of the framework is the algorithm, which has to combine the tags, weights and a matrix of values into a single similarity score.

The framework provides the mechanism to ensure that the model can be commutative, even where the algorithm is not because there is a grouper present, by reversing the presentation of the sentence pair thus giving the algorithm module two matrices. In practice, this second step will not be required by a sentence similarity model using the framework and would represent an unnecessary computational overhead.

The algorithm is divided into three modules: Word Interaction Similarity; Topic Similarity; and Combiner. The first two modules directly correspond to the Linguistic principles upon which the framework is based.

The topic similarity module needs a function to combine similarity scores and weights into a single number. It uses this function to compare the meanings of the clauses of the two sentences. Then each clause comparison is assigned a weight and the function is used again to reduce the clause similarities into a single value.

For the two subject clauses each word could have a similarity given below it in ():

The	ginger	cats
(0.03)	(0.02)	(0.67)

My	dog
(0.03)	(0.67)

The topic would combine these further to a single value for the clause comparison with the weights, in this example the value may be around 0.55 due to the influence of the noun carrying greater weight.

The same process could be repeated for the like-for-like clause comparison giving a score for each of the three clauses. This is performed for both sentence 1 compared against sentence 2 and vice-versa.

Sentence1 to Sentence 2 => (0.55) (0.67) (0.10)

Sentence 2 to Sentence 1 => (0.10) (0.67) (0.55)

Topic might come out from its algorithm as: 0.51.

The Word Interaction similarity module must determine how the sentences functionally relate to each other and how the meanings combine in groups. Each clause needs to be categorised with respect to its function towards the meaning. This needs the clauses' relative arrangements to be compared as a single measure. Each clause is given a similarity score and this is weighted. The weighted scores need to be combined to take into account

the arrangement, which can be judged using the verb clause and the other clauses position with respect to the verb clause.

While both sentences had the same basic clauses directly compared to one another, the very weak comparison between the object clauses means that it is unmatched. So we get a word interaction vector of values that might look like this:

$$w_i = (1.0) (1.0) (0.0)$$

The word interaction could then be combined into a single value, with the verb clause being more significant than the object clause, and could combine to value of 0.75.

The combiner simply has to combine the topic and interaction scores into one value. In its most complex form, the method of combination depends upon the classification of the sentence. In the more general case, at low values of topic similarity, it is the topic that dominates the overall similarity, but at high values, it is the word interaction that needs to dominate.

The combiner returns the output for the whole model. It simply combines the word interaction similarity and topic similarity as single values. So the 0.51 and 0.75 might combine as a mean to give 0.63 for the models output.

4.6 Module Development

The need to include general Linguistic concepts leads to some of the modules being necessarily flexible in their potential implementation, giving only a stated purpose of each the modules. A clearer picture of each module should become apparent with the implementation and the gradual incorporation of Linguistic concepts to the model.

Chapter 6 will take the framework and constructs a working mathematical sentence similarity model.

In chapter 7, a closer examination of the context stage is made examining the

disambiguation weights.

Chapter 8 develops the parser and word interaction modules to handle clause data.

Chapter 9 focuses on the word meaning similarity module and knowledge base.

Chapter 10 further develops the word interaction and topic similarity modules.

Chapter 14 alters some core modules including the combiner in order to handle opposites.

Although the framework outlines the structure and function of each module, the order of calculation of an individual implementation may not be the same as shown by the framework. Instead, the model can be expressed to conform to the framework, but in practice it can be inefficient to calculate the data as shown by the framework due to unnecessary duplication or overlapping of ideas.

4.7 Conclusions

This chapter presented the framework that is the key to realising a Linguistic approach for sentence similarity. Through the use of the Linguistic concepts outlined in section 2.2, the architecture for an extensible modular framework was developed. The framework is a fundamental component to meeting the objective laid out in section 1.4 and allows the experimental method to be followed with a gradual improvement of sentence similarity in the next chapter. The full power of the framework should become clearer as the implementation of the sentence similarity models using the framework develops from chapter 6 onwards.

The framework enables a sentence similarity model that adheres to it, to gradually have Linguistic concepts added to through the development of individual modules, which is necessary to conform to the experimental methodology declared in the next chapter. This chapter defined the constraints upon each module in the framework, in terms of its output and function together with its objective in preparation for their implementation.

The purpose of the framework was to provide a common structure for evolving a sentence similarity model in a manner that could test the effect of individual concepts. This purpose is accomplished and will be demonstrated from the subsequent chapters.

However, the framework through its definitions using general concepts based upon Linguistics, also has potential to incorporate ideas beyond those used in the experiments in this thesis. For example, the grouper module is not implemented as part of this research.

5.0 Experimental Method and Evaluation

5.1 Introduction

This chapter explains the experimental method and how the premise of the objective to show that Linguistic concepts could be used to produce a more accurate sentence similarity model, will be tested and evaluated.

The modular Linguistic framework described in the previous chapter is the foundation for the experimental method and allows for comparison between the effect of adding a single Linguistic concept to a sophisticated model that adheres to the framework.

This chapter details the purpose of the experiments and explains how the experiments are broken into two sections. First there is the core experiment where the main interest is in the relative performance of a model with, without an implementation of a Linguistic component. The second part evaluates the sentence similarity model on specific domains.

5.2 Purpose

There are too many Linguistic ideas to realistically include them all in a sentence similarity model. Instead, the purpose of this investigation is to focus on the core concepts that relate to the general task of sentence similarity (section 2.2). The experimental objective is to determine whether a sentence similarity model, including a particular linguistic concept, is superior as a result of the concept's inclusion.

A superior model is not guaranteed to give a more accurate measure of similarity for every possible sentence pair. Instead, it is judged to be superior because it has a higher probability of giving a more accurate answer for an unknown arbitrary sentence pair. In other words there are more situations than not, where the superior model will give a more

accurate answer over the inferior model.

Any function $h(u, v)$ can be expressed as a function with an extra parameter 'w' to give a function $g(u, v, w)$ which is invariant with respect to 'w,' so that the output remains the same as that of the original function $h(u, v)$.

So if $R(u, v, w, x)$ were to exactly represent reality then there must exist a function with the form $g(u, v, w)$ which is capable of giving the same or better performance of $h(u, v)$.

Linguistics provides the mechanism for understanding language, but from a human perspective. This can be very different from an approach needed in order to achieve the same effect for a computer, which requires simple rules and mathematical expressions.

The aim is not to rate the sentence similarity as a general performance (since this would always depend upon the specific domain and idiolect of the input) but to show that the model is improved.

So while it would be axiomatic to expect the inclusion of a Linguistic concept to improve the performance of a sentence similarity model, a method is required to evaluate whether a particular concept and its implementation is leading to an improved sentence similarity measure.

By taking two sentence similarity models which are identical, one model to include an implementation of a specific Linguistic concept and the other not, it would be possible to evaluate the effect of the Linguistic concept in isolation.

The performance of both models can then be assessed by examining their accuracy upon the same dataset. Then if the version with the included Linguistic concept gives statistically significant better accuracy then it can be declared as superior by demonstrating that the inclusion of the Linguistic component had improved the model.

5.3 Summary Metrics

The results, where appropriate, will be presented in tabular and graphical form for all of the similarity scores obtained. Lines are included on the graphs of sentence similarity to join the points (representing the similarity score for each sentence pair) purely as a visual aid but have no meaning themselves.

It is useful to describe the overall performance of a model compared to the means of human rating of similarity for a dataset as a single metric. There are three standard metrics which can be used to judge the performance of the model to the human data:

Pearson's correlation function - *PCF*

Spearman's rank correlation function - *SRC*

Root Mean Square Error - *RMS*

The RMS assumes that the values from the human data are precise and simply expresses how far apart the actual output of the model was from the desired point. SRC expresses how the relative order of the output corresponds to that of the original and shows no difference in values.

Pearson's correlation is the most widely used in the literature and the most useful as it judges the shape of the graph and not only the relative order of the points. It gives a value in the range from -1 to 1 with the higher number being closer correlation.

Two other metrics are needed for evaluating the performance on classified data (where each sentence pair is categorised as opposed to being given a similarity score) such as paraphrases in the MSRP dataset (Dolan et al., 2004):

These metrics are the f-measure and Classification Accuracy. The f-measure is designed for information retrieval and combines precision (the ratio of the selected items that have been correctly classified) and recall (the ratio of items select compared to those which should have been selected) using the harmonic mean:

$$f\text{-measure} = 2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall})$$

[5.1]

The classification accuracy judges what percentage or ratio of the input sentences have been classified to agree with the desired results as a proportion of all the inputs.

5.4 Experimental Method

It is the aim of this research to introduce and evaluate a variety of general Linguistic concepts to a sentence similarity model from disambiguation through to advanced word interaction.

For the gradual evolution of a sentence similarity model to include all the concepts to be investigated requires an adaptable common framework which can be applied for all versions. The modular framework detailed in the last chapter is used throughout this thesis and based upon the Linguistic ideas expressed in section 2.2.

Starting from a mathematical model (where every word is treated the same irrespective of Linguistic function) as a baseline, increasingly more sophisticated Linguistic concepts will be added one at a time and then evaluated.

The mathematical model re-uses some established features from existing sentence similarity models to provide a solidly performing mathematical model which is then benchmarked against other models in the literature.

5.4.1 Core Experimentation

The framework allows for only the relative effect from the latest model to dominate the performance of the model. It is also the case that there are no parameters that need to be tuned from a dataset. Therefore, it was possible to use a single smaller dataset throughout the core experiments. The ten pairs dataset (section 3.5) was used because it contained

sufficient Linguistic complexity and variation for evaluation of the effects from the changes of adding specific Linguistic features to the model.

After creating a novel mathematical model that adhered to the framework, a computational implementation that emulates the key component of the Linguistic concept being investigated, is added to the model at each stage. This was developed independently of the dataset with no tuning of parameters which could risk leading to over-fitting.

Since the aim is simply to show each concept can be adapted and benefit the accuracy of a sentence similarity model, it does not matter if it is suboptimal, as long as it is providing the core functionality of the concept. In some instances, the limit from an idealised version of the implementation of a concept is tested, using human tagging replicating the desired task. This is equivalent to possible input to the sentence similarity model where a person selects the meaning from a list or a part of speech, however, this is not the primary manner that sentences would be input to a sentence similarity model where there would be no additional information.

5.4.2 Statistical Significance

The Pearson's correlation will be taken as the primary summary metric as it includes both the relative position of the output and its magnitude.

There is no fixed method of determining statistical significance for correlations. It would be even possible to regard any improvement in a correlation function within the precision of the reported results (0.01), as indicating significant improvement since it is already a summary metric and already considers many factors.

An improvement in Pearson's correlation to the number of significant figures present in the literature (0.01) has been implied to represent an improvement between models for the STASIS-30 (O'Shea et al., 2008) dataset. Several of these models have involved tuned parameters (i.e. DTW (Liu et al., 2007)) further potentially raising the chance of anomaly.

While there could potentially be significant noise on the overall value, the test for the core experiment is whether the relative improvement from an individual factor has improved the model. Most of the noise being added to the system from the dataset and model will remain constant, so an improvement in the relative correlation can only come from the latest implementation, dramatically reducing the size of change that could occur from noise.

There would still remain additional noise from the specific additional component's implementation. The use of a stringent threshold of 0.05 PCF increase was decided to be treated as significant in light of the smaller size of the dataset.

If the implementation were to have used any tuned parameters dependent upon the dataset, then it would not be a large enough sample to be used.

As the experimentation is progressively adding new components to the sentence similarity model any improvement could be accumulative which would further reduce the odds of the noise randomly moving the answer closer to the human scores for successive versions. Any noise increasing the correlation would make it less likely that future noise would act in the same direction.

Statistically significant is only an indication that it is highly likely to be a significant result, but it would not mean it was impossible that the result obtained was an anomaly. The principles that are being added are from Linguistics which allows for further analysis to potentially identify the significant contributions to the noise from the dataset or human scores. A combination of experiments with idealised human knowledge and further discussion, greatly increases the possibility that any false positive result would be highlighted.

Finally, a more detailed examination of the results is made with respect to highlighting any potential anomalies in this assumption that might have occurred in the experimental results.

5.4.3 Benchmarking

The most advanced sentence similarity model at the end of the core experimentation will be benchmarked on the larger thirty pairs dataset. This was to allow comparison between models with very different architectures. As there are many differences that can add noise to the results, a larger dataset allowed for reducing the noise within the dataset. This therefore provides a method of determining consistency between the two datasets and provides a potentially better benchmark prior to the domain testing. It also functions as further validation that the performance was maintained on the expanded thirty pairs dataset.

5.4.4 Domain Testing

After the core testing, the resultant model will be considered for its overall performance as a similarity model in two domains. Firstly, for the task of paraphrase identification with the MSRP. This allows for the model to be compared against a wide range of specialist applications. Secondly, for opposite datasets.

5.4.5 Opposites

The final development of the model will introduce the idea of opposites which alters the similarity metric and requires a specialised domain, but it can still function as a final validation of the sentence similarity model's performance from including Linguistic concepts.

5.4.6 Conclusions

This chapter detailed the core experimental method and detailed how the datasets from the previous chapter would be used for evaluation. Description of the summary metrics and

how the performance between models was going to be determined was also included as an important step, prior to the start of the experimentation. There was also a discussion as to how the noise was being limited for the experiments.

The experimental methods were essential to test whether the objectives set out in section 1.4, have been included in this chapter. Together with the framework, a stage has been reached where the core development can begin.

The next chapter will present the framework, which is the foundation of the sentence similarity model that will be extended with Linguistic concepts and evaluated using the approaches described in this chapter.

6.0 Mathematical Model

6.1 Introduction

The last chapter set out the core experimental approach. The first stage of the experiments is the creation of a mathematical sentence similarity model that adheres to the Linguistic framework (chapter 4). While the framework is designed to allow the introduction of the Linguistic components, the objective is to show that Linguistic components could improve sentence similarity.

The core experiment required a non-Linguistic mathematical model to which the Linguistic components could be added later. This model is being described as a mathematical model because it treats all of the words consistently, without any specific consideration of the words' Linguistic function. However, several of the stages contained within the model's adherence to the framework still correspond to a Linguistic element.

This chapter presents a novel mathematical sentence similarity model called SARUMAN. While it re-uses many components present in existing sentence similarity models, it uses a much purer algorithm for combining the word similarity scores in to the single measure needed for sentence similarity.

SARUMAN is first compared against the existing models using the standard dataset in the literature, STASIS 30 (O' Shea et al., 2007). Although, as pointed out in chapter 3, that STASIS-30 dataset was inadequate for evaluating models using Linguistic features as they were all definitions with no variation in the verb clause, it is possible to test SARUMAN as it is yet to include Linguistic features. Then the ten pairs dataset is used to provide the starting point of the core experiments.

6.2 Implementation

To produce a working mathematical sentence similarity model that conforms to the framework, it is necessary to define each of the modules. Several of the modules needed can take advantage of earlier research and reuse features that are proven as part of existing sentence similarity. The standard similarity range is used of 0 to 1.

The first idea to consider, is that of a null module. The output of a null module is such that it has no overall effect and returns its input unaltered, as its output. This is the case with the grouper module, which is not included as part of the experimental development of the sentence similarity model in this thesis. The purpose for including null modules is to allow them to be replaced in later versions.

It is also the case that a mathematical model does not use the output of the parser and so this module can be set as a null module. So while the mathematical model is built to use the framework, it does not use most of Linguistic features but represents a basic implementation to be the foundation of the core experiments.

6.3 Knowledge Source

The first step that is required is to provide the source of the knowledge which forms the basis of the comparison. WordNet (Feldbaum (ed.), 1998) was selected as the knowledge database in order to provide the structure for each meaning which can be used with the word meaning similarity module.

The ontological structure within WordNet differs considerably between its four databases for the main parts of speech (adjectives, adverbs, nouns and verbs).

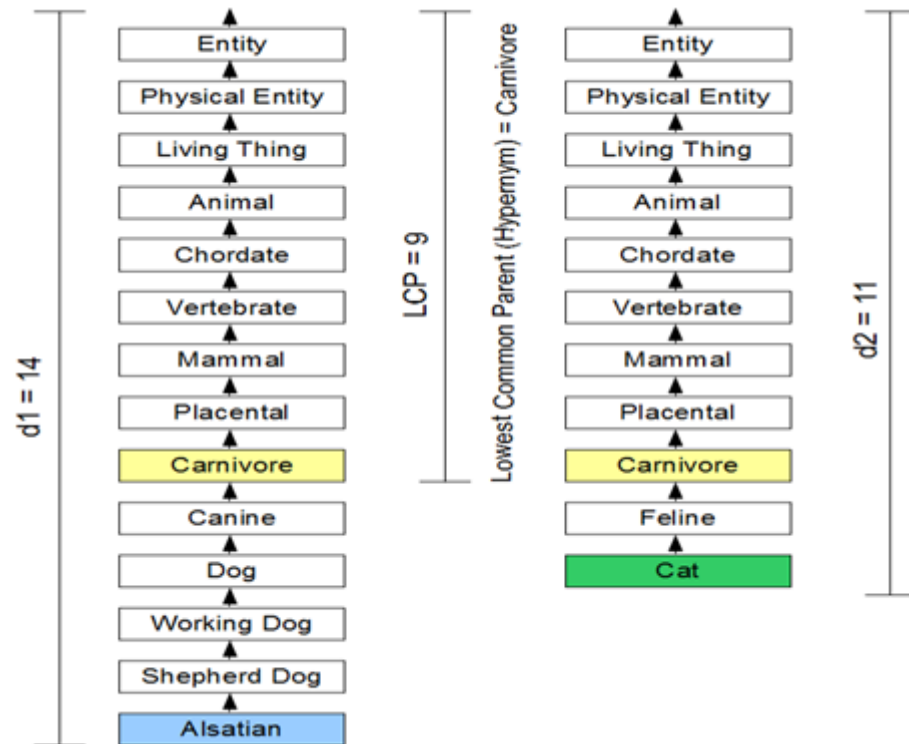


Figure 6.1: WordNet hypernym chains for “Cat” and “Alsatian”.

Each noun has a single parent hypernym which allows for a convenient chain meaning structure to be formed. Figure 6.1 shows the hypernym chains for a pair of meanings for the words 'cat' and 'Alsatian'. This produces a lowest common hypernym of 'carnivore'. This allows for the three parameters needed for the word meaning similarity module to be found (section 5.1). The lowest common hypernym depth (*LCH*) can function in equation [5.2] as the common structure, *C12*, and the depth of each of the head words beyond the *LCH*, gives *D1* & *D2*. From figure 6.1 the depth of cat is 11 and Alsatian is 14, with a common parent of carnivore which has depth 9. So *LCH* would be 9, *D1* would be 5 and *D2* would be 2.

6.3.1 Encoding Meaning Structure from WordNet

Rather than using a stemmer and iterative pointer look-up to access the WordNet data it was instead all extracted for direct use. Each stem word is reversed stemmed for its type to give all of the valid forms for that stem. So "love" as a verb would give "loved", "loves", "loving" and "love" as its valid forms. Each meaning is then represented by a string of

characters each representing a synonym group. Duplicates of characters can occur representing different groups as long as no child nodes duplicate. A question mark is reserved as a special character to show an unknown group.

So "animals" could become represented by a string like "n121" where the 'n' = "entity" the first '1' = "physical entity", the '2' as "living entity" and the final '1' as "animal".

Through, using a single representation of the meaning, it avoids multiple unnecessary searches of the WordNet databases. In addition it allows the exclusion of the unused information within WordNet and so is therefore faster to execute.

Although the hypernym chain should be unique for all words, there are a few rare anomalies such as "wheeled vehicle" which is given two alternative parents of "container" and "vehicle". This is resolved by splitting the meaning into two meanings. So one chain for "wheeled vehicle"-"container" and another for "wheeled vehicle"-"vehicle".

The ontological structure within WordNet is richest for the nouns (WordNet, 2009). While the verbs contain the hypernym relationship (also described as troponyms), adjectives and adverbs have a completely flat ontological structure within the database.

Some adjectives are given an ontological link to a noun and this structure can be used to represent the meanings of the adjectives. An additional node is added as a root node to represent an adjective, then the noun ontological chain is added to this root node to give an ontological structure equivalent to that of the nouns. So for example "an123". For adjectives with no links a terminal node of '?' is added to give "a?".

For the adverbs, it is necessary to include a further ontological relationship to first identify a related adjective, since there are no links to nouns. There could then be a chain "ran123".

The verb structure lacks a single common root node in the way that nouns had "entity". However, it would be possible to view there being a common verb: "to do". The verb tree is much broader and shallower than the nouns, with "to make" being given 49 definitions for example. Some of which are indistinguishable from each other but connected to distinct hyponyms. Therefore, a second layer is also added to group the different root nodes

together. To some extent this knowledge is already contained in the WordNet application (WordNet Documentation, 2009) where labels such as "verb stative" are added.

The verbs are encoded like adjectives when there is a linked noun but with a different leading character ("vn1257") alongside its verb structure, such as "vs124" with the 's' in this case being "exists".

This encoding allows for faster access and simple processing. To find the depth of the lowest common hypernym, both strings are iterated until the characters do not match and the length minus this distance gives the distinct meanings.

6.4 Word Meaning Similarity Module

Li et al. (2003) examined a large number of possible word algorithms using the *LCH* and the distance between the two meanings being compared. This is the same algorithm as was used by STASIS (Li et al., 2006) with the values of $\alpha = 0.45$ and $\beta = 0.2$ described as optimal.

This gives the following formula in terms of $C12$, $D1$ and $D2$:

$$\tanh (0.45 * C12) * \exp (-0.2 * (D1 + D2))$$

[6.1]

where

$C12$ = length of matching string until first difference (*LCH*)

$D1$ = "length of meaning 1" - $C12$

$D2$ = "length of meaning 2" - $C12$

The Li et al. (2003) formula does not give unity for comparing a meaning against itself, so

a final condition is added in case the forms are identical, with identical meaning then the similarity is the maximum of 1.0.

6.5 Weights

While not a module that is required for the model to function, an importance weight is added using Resnik information weights (Resnik, 1995), based on the frequency of occurrence of words in the corpus.

$$\text{Importance Weight (w1 \& w2)} = \log(n1) / \log(N+1) * \log(n2) / \log(N+1)$$

[6.2]

Where 'n1' is the number of times word 'w1' appears in the corpus of and 'N' the total number of words in the corpus,

Brown's corpus (Francis and Kucera, 1979) is a long standing indexed corpus of English language which is suitable for use with Resnik (1995) weights and is the same corpus as used by STASIS (Li et al., 2006). This means the minimisation of the variation between the mathematical model from the knowledge sources.

Brown's corpus (Francis and Kucera, 1979) is used to estimate the relative frequency of each word. Any word that is not present in the corpus takes a frequency of 0 and so returns a weight of 1.0.

Disambiguation weights that are needed in order to eliminate unwanted possible meanings of each form (see section 5.3) are not used for the basic module and disambiguation is based purely upon the context of similarity. Every meaning is set as a disambiguation weight of 1.0 which means all meanings are treated as equally valid.

6.6 SARUMAN Algorithm

The next critical stage is combining the matrix of meaning scores and weights in order to give a single meaningful number for the similarity. A new algorithm is used to combine this information. The algorithm is where the model gets its name, SARUMAN standing for Semantic Analysis R(U)oot Means All as Nouns.

The matrix is reduced to a vector with a value for each word. The highest scoring valid meaning (in this case all meanings are valid) is selected for each word, giving a value and the position of the matched word in the other sentence is also noted. This gives three vectors for each sentence: the weights, the word similarity score and the matched positions. All the vectors are the same length as the number of words in the sentence.

Where there is a similarity score of 0, the position is set to -1 and where two words identically give the same meaning then the shorter distance (to be defined shortly) is used.

If all of the values are on the same scale then the simplest way to construct a single representative value is to find the weighted mean:

$$\mu(x) = (1 / \sum(w[i])) * \sum(x[i] * w[i])$$

$$\text{where } 0 \leq w[i] \leq 1.$$

[6.3]

The weighted mean is unaffected by adding a new value $x[i]$ when the weight $w[i]$ is 0.

Formula [6.3] is not necessarily commutative if only the contribution from the words in one sentence were used, which is a requirement of the framework. Accordingly, the weighted mean is found for the contribution from each sentence. ' μ_1 ' is the weighted mean for sentence 1 compared against sentence 2 and is multiplied by ' μ_2 ' when the sentences are reversed (sentence 2 compared against sentence 1).

However, the individual contributions of all the words from just one sentence would account for the total topic similarity. Therefore, multiplying the weighted means for both sentences has altered the dimensionality. To restore the dimensionality to the topic similarity, the square root is taken so that:

$$S_{topic} = \text{sqrt}(\mu_1 * \mu_2)$$

[6.4]

A simple version of the word interaction similarity module is used and this only uses the word order and ignores other word interactions. First, a description of the distance between a pair of matched words in the two vectors needs to be defined.

The length of the sentence is taken as the distance between the first and last point. Every point represents the position of a word. If the length of any sentence is taken to be unity with all the words equally spaced then, for a sentence of n words, the distance along the line of the j^{th} word becomes:

$$\begin{aligned} \text{Distance}(j, n) &= (j - 1) / n - 1 & \text{for } n > 1 \\ \text{Distance}(j, n) &= 0.5 & \text{for } n \leq 1 \end{aligned}$$

[6.5]

The words' position in the sentence can be thought of as evenly spaced points on a line of length 1. With the first word in the sentence being the left most point. For $n = 1$, it is a special case and the word is placed in the middle of the line maintaining the total length of the line as 1, hence the distance is 0.5.

The proximity of a sentence needs to also be defined. If the maximum similarity score for a word is below a threshold (0.2), then the word is considered unpaired.

A threshold is used to reflect the fact that not all words have a corresponding match in the other sentence. If a word has only very weak similarity with words in the other sentence then it is not clear that it has a relative position to calculate. The value of 0.2 means that pairs with a low similarity will be excluded and is the same as used by STASIS (Li et al., 2006).

If a word is unpaired, then the proximity is taken as the minimum, 0.0, otherwise the following formula is used:

$$Proximity(j, k) = 1 - |distance(j, k)|$$
[6.6]

Using scaled proximity for each word pair with a weight then the position similarity becomes:

$$S_{interaction} = \sqrt{\mu_{p1} * \mu_{p2}}$$
[6.7]

The combiner module is simply defined as a simple linear function:

$$S_{sentence} = S_{topic} * ratio + S_{interaction} * (1 - ratio)$$
[6.8]

Several of the modules' implementations duplicate the function of STASIS and so to avoid the need for tuning the same value of the ratio is used as for STASIS of 0.85 (Li et al., 2006).

6.7 Experiments

A version of SARUMAN that includes the modules as defined in sections 6.2-6.6 was implemented together with an extracted version of WordNet to provide the vocabulary for the knowledge source. SARUMAN was then run on each of the sentence pairs in the STASIS-30 dataset.

In addition a more primitive version of SARUMAN was also tested where the Resnik (1995) importance weights were excluded. Effectively removing equation [6.2]. This experiment was included because it would have been possible to use a simplified version of SARUMAN as a benchmark and still adhere to the framework. Resnik weights were quite a substantial component and their influence on the model is significant. However, the purpose was to start from a significant mathematical model so they had been included.

An implementation of STASIS (Li et al., 2006) to use the same the version of WordNet as SARUMAN was also used for the experiments rather than directly using the values in the literature. SARUMAN shared many features from STASIS including the Li et al. (2003) word similarity algorithm so as close a version as possible could be compared.

The values for LSA (Deerwester et al., 1990) were taken from its online implementation (LSA, 2013) and the other main models (section 2.9 & 2.10) with values in the literature are also compared side-by-side to SARUMAN in graphical form. The other models displayed are: OMIOTIS (Tsatsaronis et al., 2010), IISIS (Islam and Inkpen, 2008), SyMSS (Oliva, 2011) and DTW (Liu et al., 2007).

Finally, SARUMAN is tested on the ten pairs dataset to give the reference point for the start of the core experiment (described in section 5.4.1). Only the models with official web implementations at the time of the experiment and STASIS, were compared against SARUMAN for the ten pairs dataset.

6.8 Benchmarking SARUMAN

Having constructed SARUMAN to use the framework, it would be possible to start the core experiment. However, the starting point is still a mathematical model and before testing it with the ten pairs dataset, it can be directly compared to the other models using the STASIS-30 (O'Shea, et al., 2008) (section 3.2) which has become the standard benchmark dataset in the literature for sentence similarity.

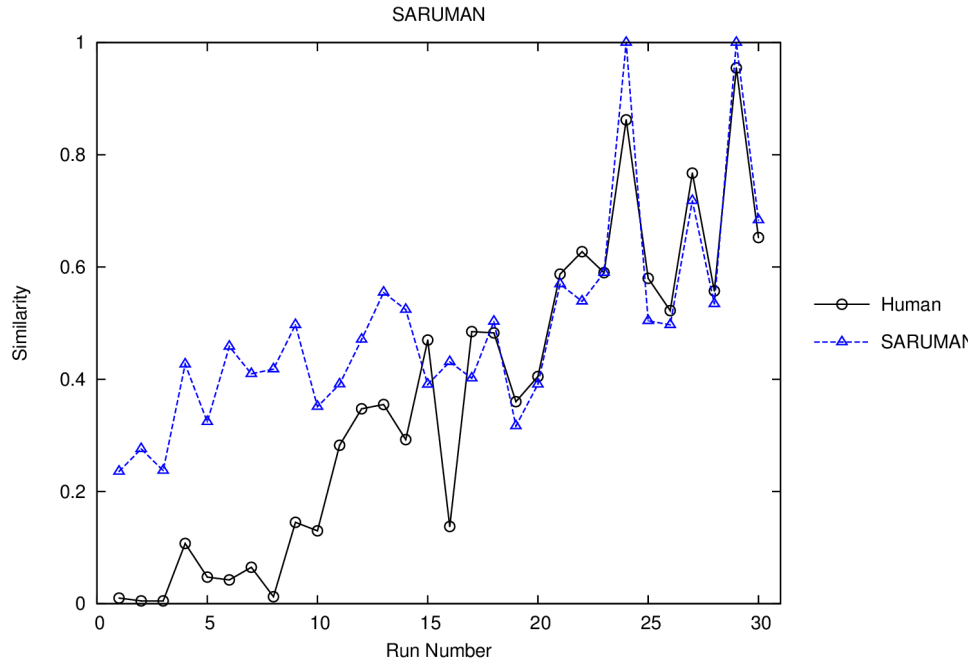


Figure 6.2: SARUMAN for the O'Shea et al., (2008) STASIS-30 dataset

From figure 6.2, it can be seen that SARUMAN performs strongly at the higher end of the human similarity scores (runs 20 – 30) but does considerably worse on the lower end of the similarity (runs 1 – 10) where it is above the human scores. Overall it gave a good Pearson's correlation of 0.820 and Spearman's of 0.793.

It was stated that the inclusion of importance weights was not essential for SARUMAN to conform to the framework and a primitive version of SARUMAN was run without importance weights to show their effect upon the output. It can be clearly seen from figure 6.3 by eye, that the Resnik weights (Resnik, 1995) are moving the output closer to the human scores and this visual rating is confirmed by the correlations with the primitive unweighted version giving Pearson's correlation (PCF) of 0.749, Spearman's of 0.699. The primitive version of SARUMAN is consistently higher in its rating and hence overestimating the similarity worse than SARUMAN with the weights.

This clearly shows that the corpus and importance weights are contributing a statistically significant improvement to SARUMAN and hence will remain included as an intrinsic part of the mathematical model.

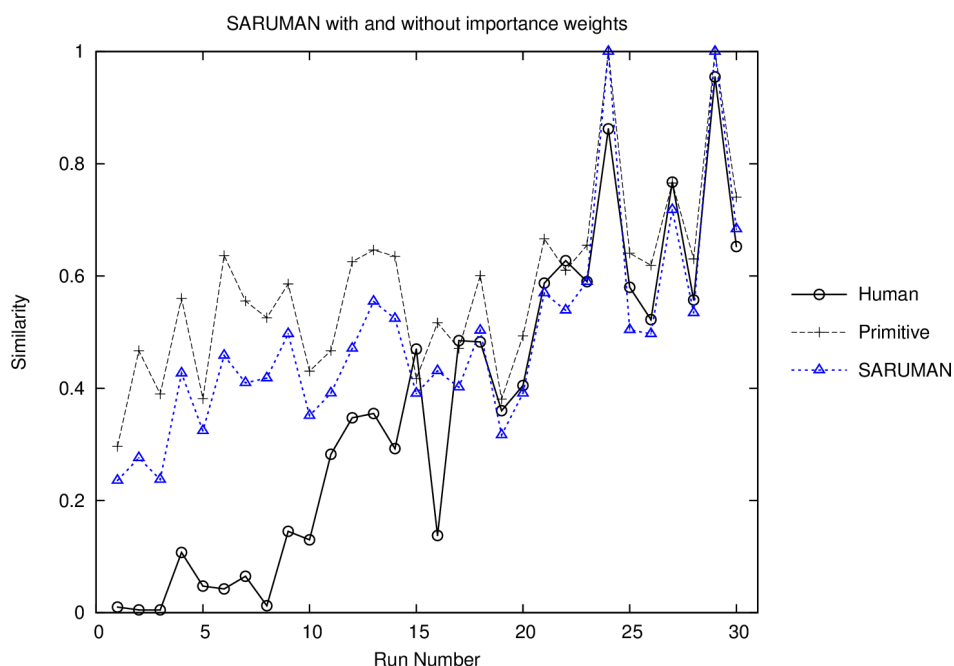


Figure 6.3: SARUMAN primitive (without weights) vs. SARUMAN

6.8.1 Comparing SARUMAN to other Models

Table 6.1 gives the Summary values for SARUMAN alongside the leading sentence similarity models (earlier described in sections 2.9 & 2.10). Other models in the literature quote similar values to these for their Pearson's correlation function in the range of 0.75-0.88. Some of these involve parameters tuned on the dataset which runs a similar issue to over-fitting as noted for the word models by Zesch and Gurevych (2007).

Model	PCF	Spearman's	RMS
SARUMAN	0.820345	0.793414	0.203964
STASIS	0.857901	0.862609	0.339729
OMIOTIS	0.855668	0.890866	0.189699
IISIS	0.846635	0.830050	0.144861
DTW	0.839062	0.859657	0.164721
LSA	0.840011	0.866273	0.157053
SyMSS	0.752471	0.700623	0.201672

Table 6.1: The correlations for the leading models.

It can be seen that SARUMAN is marginally worse than the leading models on the Pearson's correlation (PCF) with STASIS obtaining the highest PCF of 0.858 and similarly worse with the Spearman's Rank correlation (SRC) where OMIOTIS manages 0.891. It is not conclusive that the differences are statistically significant with differences in Pearson's correlation of under 0.04 and it would be fair to regard SARUMAN as competitive with the other models but fractionally worse.

The values for DTW (sentence similarity with dynamic time warping) have been adjusted from that quote, (Liu et al., 2007) for sentence pair 24 "cock - rooster" to a value of 1. The value given looked like a transcription error inconsistent with their quoted correlation and the algorithm would be expected to give close to unity for the synonymous pairs, with this adjustment the data then corresponds exactly to their quoted correlations.

STASIS was reimplemented and using a more up-to-date version of WordNet's dictionary which explains the higher PCF than the 0.84 reported in Li et al. (2006). The results for LSA (Deerwester et al., 1990) were taken from the implementation found at the LSA Colorado website (2013). One of the steps used by LSA to alter the original cosine scale, "folding the vector space", is not clearly defined. Despite being able to achieve small negative numbers for some inputs, the output is assumed to have the range 0 to 1 and would not affect the correlation but would alter the RMS (root mean square error).

Run	Definitions		Human	LSA	STASIS	SARUMAN
1	Cord	Smile	0.01	0.02	0.383	0.236
2	Autograph	Shore	0.005	0.06	0.437	0.276
3	Asylum	Fruit	0.005	0.01	0.288	0.237
4	Boy	Rooster	0.1075	0.07	0.617	0.427
5	Coast	Forest	0.0475	0.15	0.507	0.324
6	Boy	Sage	0.425	0.06	0.601	0.459
7	Forest	Graveyard	0.065	0.19	0.518	0.410
8	Bird	Woodland	0.0125	0.01	0.627	0.418
9	Hill	Woodland	0.145	0.62	0.683	0.497
10	Magician	Oracle	0.130	0.16	0.565	0.351
11	Oracle	Sage	0.2825	0.15	0.609	0.391
12	Furnace	Stove	0.3475	0.43	0.667	0.471
13	Magician	Wizard	0.355	0.26	0.712	0.555
14	Hill	Mound	0.2925	0.08	0.659	0.524
15	Cord	String	0.47	0.35	0.650	0.391
16	Glass	Tumbler	0.1375	0.45	0.650	0.431
17	Grin	Smile	0.485	0.39	0.606	0.402
18	Serf	Slave	0.4825	0.66	0.781	0.503
19	Journey	Voyage	0.36	0.22	0.598	0.317
20	Autograph	Signature	0.405	0.40	0.657	0.391
21	Coast	Shore	0.5875	0.56	0.807	0.570
22	Forest	Woodland	0.6275	0.50	0.749	0.539
23	Implement	Tool	0.59	0.66	0.831	0.590
24	Cock	Rooster	0.8625	0.97	0.994	1.000
25	Boy	Lad	0.58	0.66	0.659	0.504
26	Cushion	Pillow	0.5225	0.26	0.667	0.497
27	Cemetery	Graveyard	0.7675	0.48	0.822	0.718
28	Automobile	Car	0.5575	0.74	0.702	0.534
29	Midday	Noon	0.955	1.00	0.996	1.000
30	Gem	Jewel	0.6525	0.72	0.829	0.684

**Table 6.2: Numerical results for SARUMAN, STASIS and LSA (LSA website,2013)
for STASIS-30 dataset (O'Shea et al., 2008)**

The numerical outputs obtained for these two models are given alongside SARUMAN in table 6.2, all other values used are those published in the literature. Figures 6.4-6.9 show the results graphically side-by-side with SARUMAN. It can be seen that STASIS, which shares several of the features in the modules of SARUMAN, is consistently further from the human rating despite its superior PCF value. The other 4 models do better at the low end of the scores with corpus methods being closer, but still demonstrate considerable

noise and not quite as good as SARUMAN was on the high end of the human similarity scores (Runs 20-30).

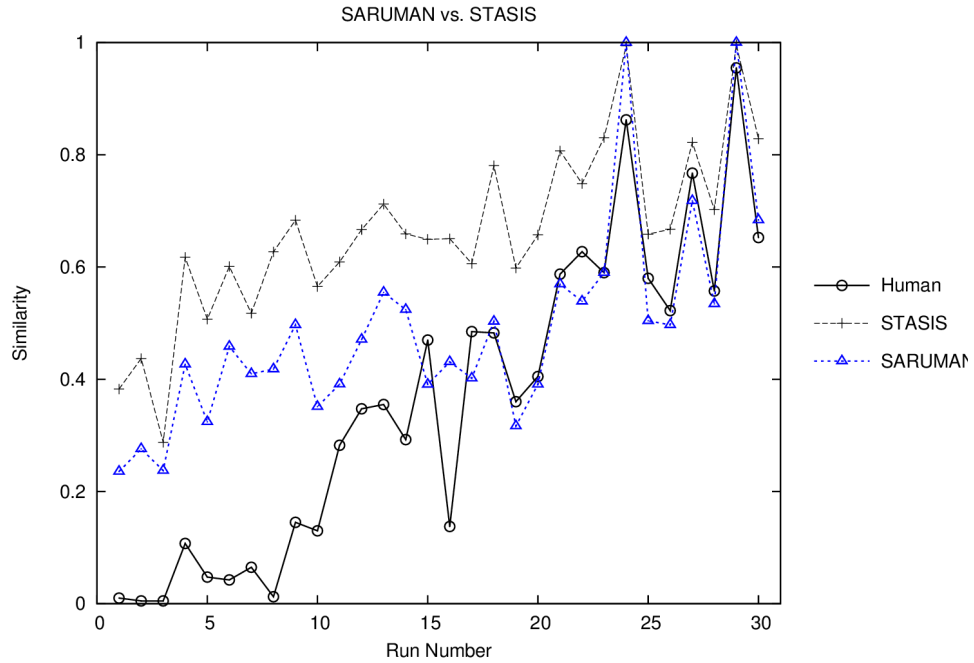


Figure 6.4: SARUMAN versus STASIS (Li et al., 2006) for the STASIS-30 dataset

In figure 6.4 STASIS (Li et al., 2006) shows a markedly worse performance than SARUMAN and it overestimates the similarity across the whole range but does include an upwards trend in the values consistent with the human ratings.

LSA (Deerwester et al., 1990) figure 6.5, shows results much closer to the trends in the human data. It does include points which significantly differ in all parts of the range, (especially runs 9, 14 & 21) so is slightly worse on the high end than SARUMAN.

Figure 6.6 shows that IISIS (Islam and Inkpen, 2008) can be seen to give reasonable tracking of the human scores with few runs where it is showing a substantial difference between its results and the human scores. Where it shows a weaker performance compared to SARUMAN is on the upper end of the similarity (runs 20 -30).

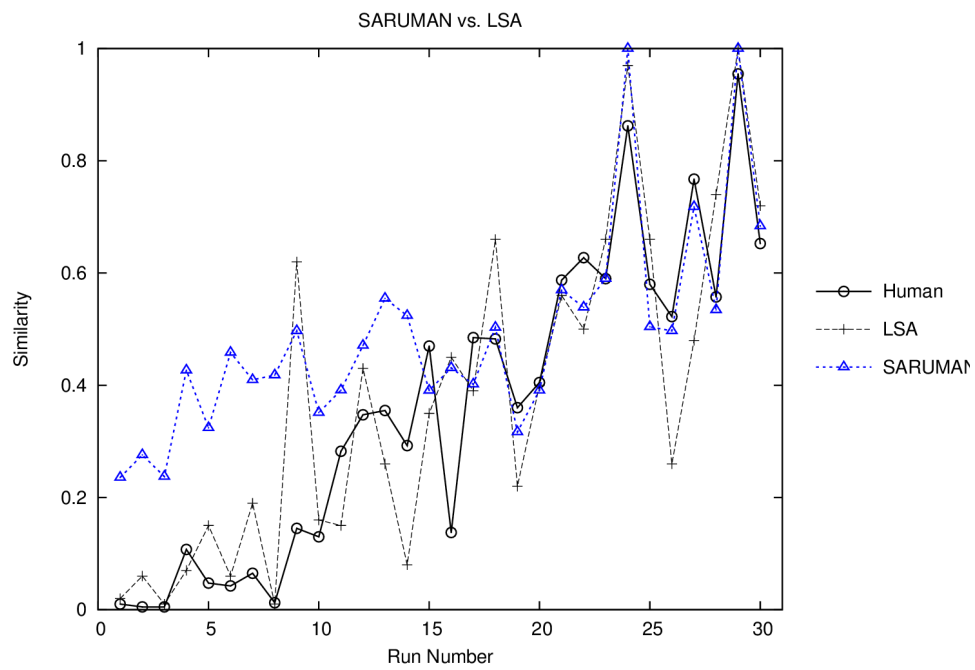


Figure 6.5: SARUMAN versus LSA (website, 2013) for STASIS-30 dataset

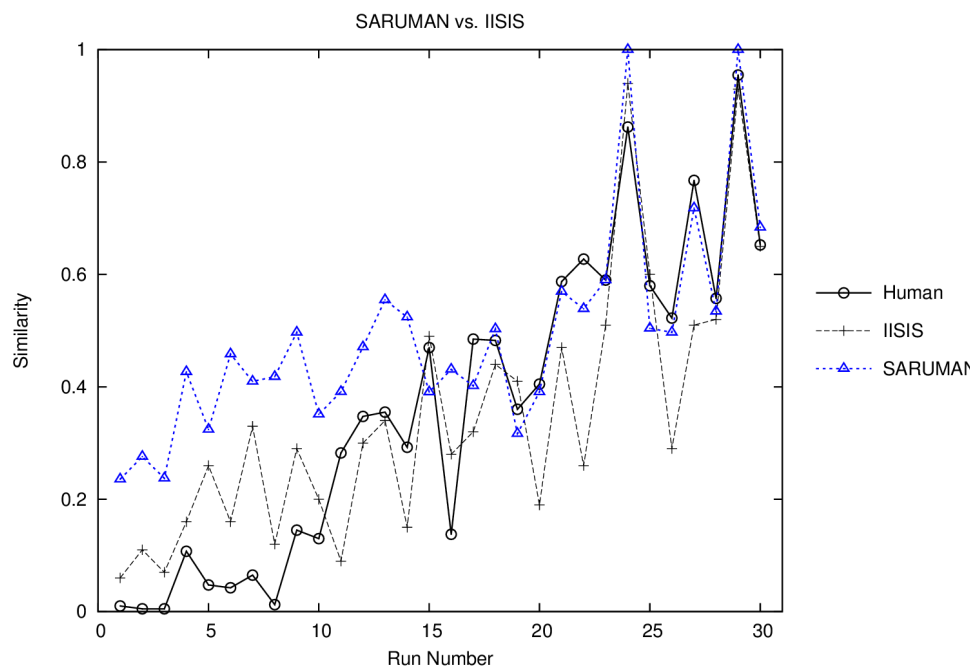


Figure 6.6: SARUMAN versus IISIS (Islam and Inkpen, 2008) for STASIS-30

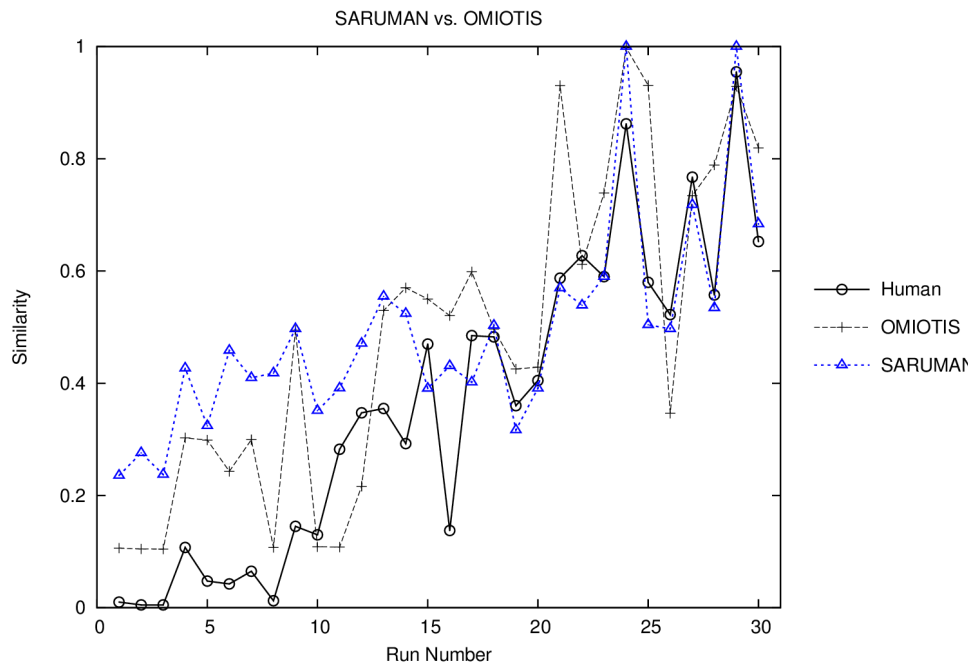


Figure 6.7: SARUMAN versus OMIOTIS (Tsatsaronis et al., 2010)

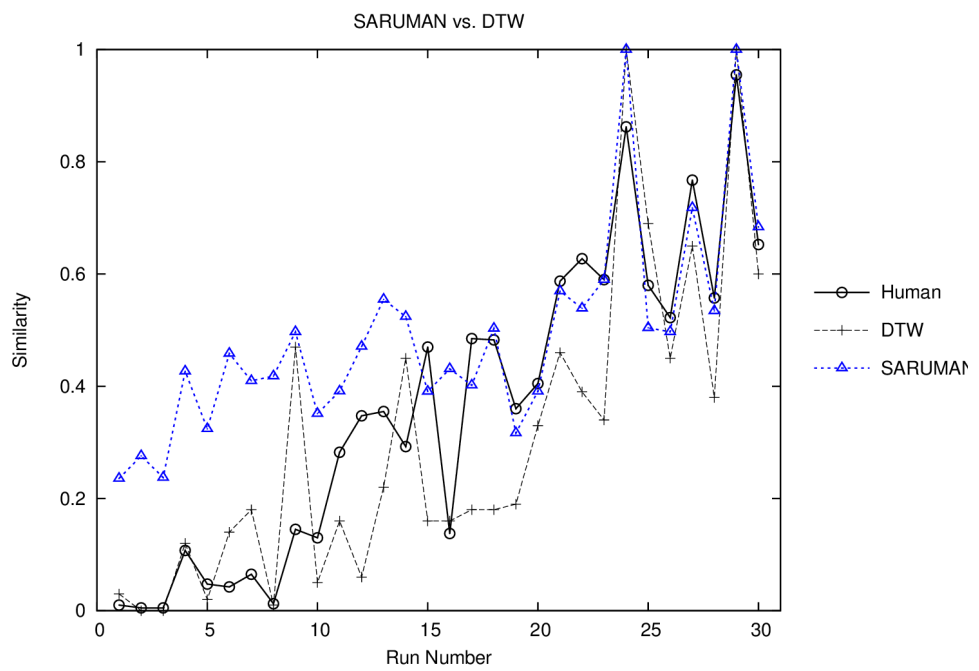


Figure 6.8: SARUMAN versus DTW (Liu et al., 2007)

From figure 6.7 it can be seen that OMIOTIS displays clearly corresponding regions for the low, medium and high human ratings. It still differs from the human scores but, despite consistently overestimating the similarity scores at the low end, is closer than SARUMAN.

In Figure 6.8, DTW shows a much closer match for human scores on the low end of similarity (runs 1 -10) than SARUMAN but often underestimates in the middle range and lacks some of the shape in the human values.

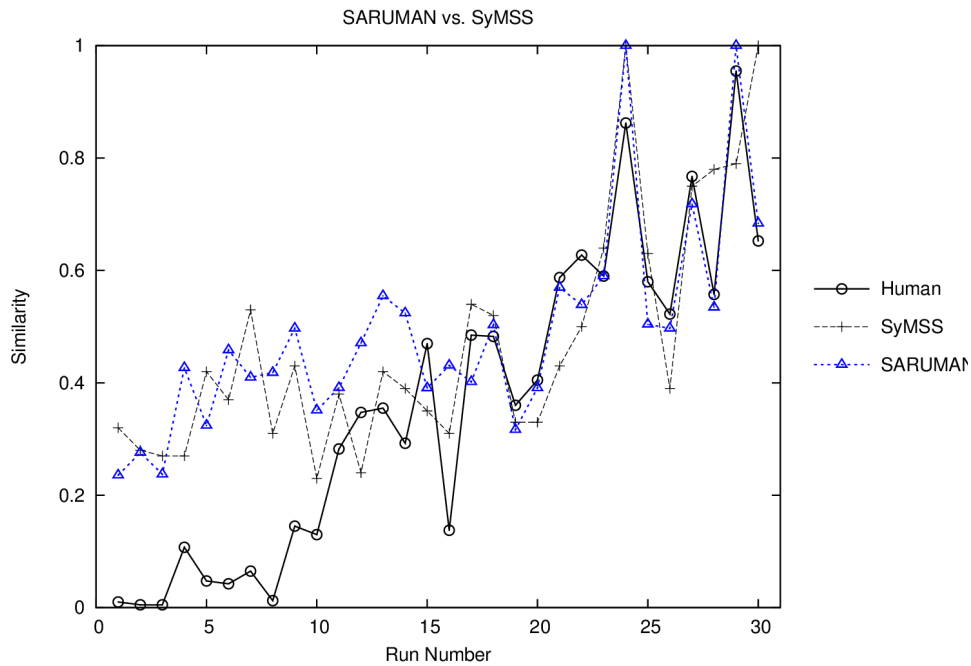


Figure 6.9: SARUMAN versus SyMSS (Oliva et al., 2011)

The final Model to be compared against SARUMAN graphically is SyMSS (Oliva et al, 2011) which is a recent model using part of speech information which is of particular interest with respect to how SARUMAN will be developed in the next chapter. Looking at figure 6.9, it can be seen that highly similar results to SARUMAN are achieved despite SyMSS showing significantly lower correlations. However, Oliva et al. (2011) report that SyMSS did better than several of their own implementations of other models for correlation. The sparseness of non-nouns makes it difficult to assess the relative impact of their inclusion of a Linguistic element, but it seems as if it had not lead to an improvement over the other models on the dataset.

Although SARUMAN shared many of its module designs with STASIS, SARUMAN's algorithm module makes it fundamentally different from STASIS. There are some other structural nuances which give greater potential for expansion but these are not relevant to its current performance, when viewed as a stand alone algorithm.

STASIS uses a bag of words formed from both sentences in order to create vectors of the same length. This was in part, a restriction imposed by its algorithm which could not handle vectors with different lengths. This was an artificial step which SARUMAN manages to exclude and can process vectors of numbers that correlate directly to the sentences. This accounts for its results being able to be much closer to the human scores.

6.9 Experimental Benchmark

Having shown that SARUMAN is performing competitively with other mathematical and sentence similarity models, the core experimentation of the thesis can commence. SARUMAN will be used to provide a baseline for the inclusion of Linguistic concepts. As stated in the experimental methodology (section 5.3), a single dataset needs to be used throughout the development and core evaluation, and that the ten pairs dataset has been chosen as it has significant Linguistic features and variation, which will need to be tested.

Since this is a new dataset, it is only possible to test using models that are available either through the internet or re-implementation. Methods that use corpus statistics are prohibitive to replicate due to the lack of availability of the same corpus. Therefore it is only STASIS that has been re-implemented and two models available through web applications LSA (Landnauer and Dumais, 1997) and OMIOTIS (Tsatsaronis et al., 2010) which have been assessed alongside SARUMAN.

Model	PCF
SARUMAN	0.245247
OMIOTIS	0.142011
LSA	0.618718
STASIS	0.123989

Table 6.3: Correlation for New Dataset

Only the results of the PCF are included in table 6.3 because the web implementation of OMIOTIS (OMIOTIS on-line implementation, 2011) was possibly using a different scale to the one published. The drop in correlation from the STASIS-30 dataset is striking, with the models that are using WordNet as their knowledge base are all struggling. There is still a positive correlation but it is weak. This indicates that SARUMAN is still struggling to cope with the Linguistic and computational complexity within the ten pairs dataset.

SARUMAN gives significantly better correlation than OMIOTIS and STASIS but is giving clearly worse performance than LSA. Figure 6.10 shows how LSA values correspond to the human values (the lines are only included for illustration purposes and possibly visually exaggerate the performance). LSA gives values which emulate the shape of the human scores but some with some of its values, LSA significantly fails to match the human scores, in particular pairs 3 and 9 (where it misses any differences in the pair).

It is probable that LSA is doing better due to fewer disambiguation issues on the dataset for the corpus method than exists using WordNet as a knowledge base (as was the case with SARUMAN, OMIOTIS and STASIS). The next chapter will be investigating how to Linguistically include disambiguation for SARUMAN.

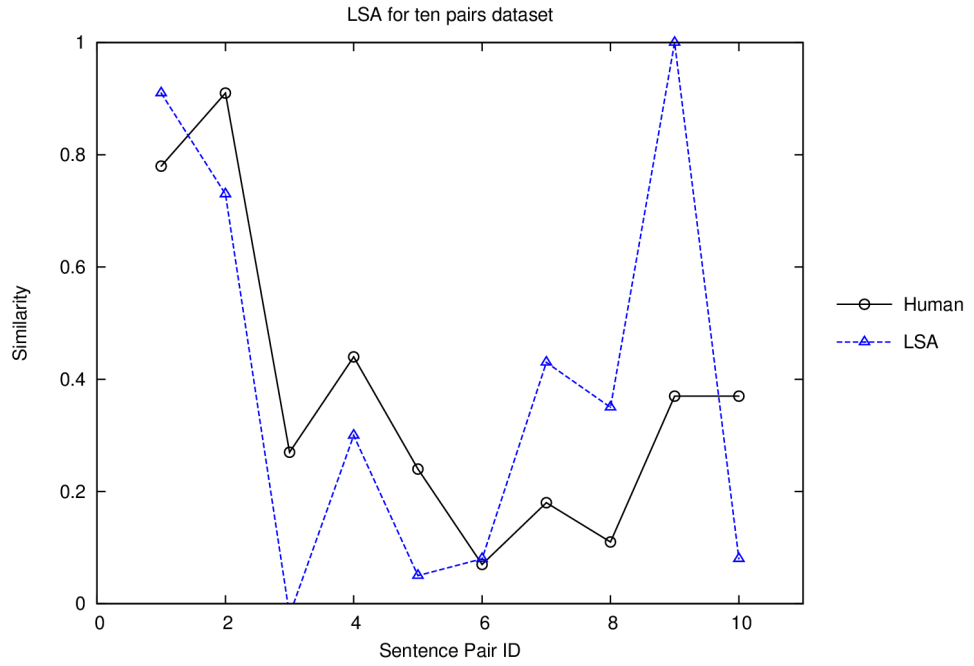


Figure 6.10: LSA with Ten Pairs dataset

Although the ten pairs dataset has highlighted some weaknesses in some of the sentence similarity models, not shown by the STASIS-30 dataset, the purpose of using the dataset was to provide a benchmark for the further experiments. Table 6.4 shows the numerical results for SARUMAN which can clearly be seen to be over-estimating several of the pairs. Its PCF is 0.283, SRC is 0.334 and a large RMS of 0.397 can be regarded as the baseline level for the later experiments on the dataset.

ID	Sentence Pair	Human	SARUMAN
1	<i>The Persian cat sat on the carpet. The ginger cat sat on the mat.</i>	0.78	0.667
2	<i>The caterpillar metamorphosed into an elegant butterfly. The caterpillar changed into a beautiful butterfly.</i>	0.91	0.833
3	<i>Fish swim in water. Birds fly in the air.</i>	0.27	0.637
4	<i>They believed the red bus was environmentally friendly. They put their faith in the train being green.</i>	0.44	0.544
5	<i>To drive a manual car, you must press down the clutch. To open the window, the mouse has to be double clicked.</i>	0.24	0.517
6	<i>The green grass glimmered as the sun shone on the morning dew. The ancient building had stood on that small hill for eons.</i>	0.07	0.440
7	<i>The Persian cat sat on the carpet. The Persian rug was on the dresser.</i>	0.18	0.792
8	<i>The exploded diagram shows how cars work. The car exploded at the art show.</i>	0.11	0.753
9	<i>Woman, without her man, is nothing. Woman: without her, man is nothing.</i>	0.37	1.000
10	<i>Trees need sunlight and water to grow. Food and drink are essential for your development.</i>	0.37	0.483

Table 6.4: Ten Pairs Dataset and SARUMAN scores

6.10 Conclusions

This chapter successfully showed that it was possible to implement a working sentence similarity model using the Linguistic framework from chapter 4. By using several existing approaches to sentence similarity, combined with a novel algorithm, it was shown that the new mathematical model is capable of competitive results on the STASIS-30 dataset.

The most important element was the production of a baseline level accuracy and a foundation upon which the core experimentation can be carried out. This baseline was showing weak but still positive correlation on the ten pairs dataset, which was higher than the other WordNet based models tested on the dataset.

The current version of the model was unable to match the performance with the corpus based LSA and gave significantly lower correlation. SARUMAN can currently be considered weaker than LSA but the purpose of the model was not to beat all extant models but to allow for incremental addition and evaluation of Linguistic features. The comparison to LSA will be returned to later in the thesis after the completion of the core experimentation, to which LSA's performance is not pertinent.

It is likely part of the reason for LSA's superior performance is down to fewer issues with disambiguation. The focus of the next chapter is introducing disambiguation to SARUMAN.

7.0 Disambiguation of Meaning and Type

7.1 Introduction

With the benchmarking of SARUMAN on the ten pairs data set, it is now possible to start adding Linguistic components to SARUMAN. The objective of the research was to show that introducing Linguistic concepts could improve a sentence similarity model.

The last chapter produced the mathematical model SARUMAN which provides the starting point for the experimentation to add in Linguistic concepts. This chapter will utilise the fact that SARUMAN is using the Linguistic framework (Chapter 4) in order to introduce the Linguistic idea of disambiguation and context to SARUMAN.

From the main experimental method given in chapter 5, a demonstration of disambiguation improving SARUMAN would require a change in Pearson's correlation of 0.05 or over from the version of SARUMAN in the last chapter.

Although SARUMAN was described as a mathematical model, it still includes approximations of several Linguistic tasks but without any consideration as to the meaning or function of the individual words.

This chapter develops the weighting module in an attempt to add context to the sentence similarity model to select a meaning for each polysemous word, as opposed to simply using the highest scoring similarity score. This is the task of disambiguation.

7.2 Disambiguation Approaches

There are two forms of disambiguation which will be investigated: selecting a single meaning; and excluding meanings that are determined to be invalid. The exclusion of the meanings will be purely based upon a word's type (part of speech) in the context of its

sentence.

While computers can perform strongly at tagging words in a sentence for their part of speech as discussed in section 2.4, full disambiguation is a task where automation will struggle to come close to human understanding. Because of the known difficulty in the automation of disambiguation, two approaches are examined. One uses human selected meanings added as tags to the data and the other a method of automatic disambiguation.

Although, an implementation of the parser module could have been used in order to automate the tagging of the part of speech, this is not done until the next chapter but is almost identical to the human tagging except on sentence 6b (see section 8.2.9). There are 4 different combinations of disambiguation which will be used with SARUMAN:

- Type disambiguation.
- Human meanings tagged to the data.
- Automatic meaning disambiguation without type disambiguation.
- Automatic meaning disambiguation with type disambiguation.

The type disambiguation will set all of the disambiguation weights to 0 for meanings other than those of the tagged type which are set to 1. Hence, excluding possible meanings for the form of the word that have a different part of speech to the tagged data. The disambiguation by meaning causes the weighting module to aim to set only a single meaning to a weight of 1, for each word.

7.3 Human Tagging

For the purpose of investigating disambiguation, it was wanted to have detailed tagging of the ten pairs data set, to give the parts of speech and meanings based upon the WordNet

(Feldbaum (ed.), 1998) definitions. To this end, two people familiar with grammar and the WordNet interface were given the data set to parse and select the meanings that they deemed closest to the intended meanings.

As opposed to the situation where collective human judgement for the similarity scores of sentences was used, the tagging of the sentences could use a single individual. This is because humans with sufficient knowledge can be regarded as experts at interpreting the intended meaning of sentences.

It is not expected that in most cases that there would be any dispute to the meaning of the sentences, since it is a requirement of effective communication that the recipient is able to glean the intended meaning. Therefore, the selection of meanings is only expected to give variation where there are very close meanings within the WordNet options. In the case where there are two meanings that are both valid in the context and have highly similar meanings, then this should be reflected within the ontological structure from where the meaning structure is obtained.

While it would have been reasonable to just use a single participant for the tagging a second individual was used to confirm that the difference between the two are small. By selecting between the cases where there was variation this becomes equivalent to the rating system used for the Microsoft Research Paraphrase dataset (section 3.4), which is one of the largest and most widely used datasets of rated sentence pairs.

While it might not be the case that the selected meanings will directly match up to the interpreted meanings of the participants, it does provide a realistic possible input of human judgement. With the assumption that the meanings of the sentences are being compared based upon a single meaning for each sentence, then the selected meanings should still give similarities in the meanings related to the similarity scores given by human judgement.

The tagging required part of speech tagging, which while should be familiar to most raters, it is not the case that all participants would have to be familiar with the basic grammatical units wanted. It is also a very time consuming process to select the nearest meaning due to WordNet occasionally having been constructed with overlapping meanings and sometimes

very specific meanings for the words.

Table 7.1 shows the human tagged version of the ten pairs dataset. Each word was tagged for its type (part of speech). nouns (nn), verbs (vb), adjectives (adj) and adverbs (adv) were given their selected meaning from the list in WordNet. The tag is followed by a hash (“#”) and a number with 1 being the first meaning in WordNet as displayed in the interface. The other types used were: article (art), preposition (pp), to, pronoun (pn) and conjunction (conj).

Sentence ID	Sentence and type tagging with WordNet meaning
1a	The Persian cat sat on the carpet.
	art, adj#1, nn#1, vb#3, pp, art, nn#1
1b	The ginger cat sat on the mat.
	art, adj#1, nn#1, vb#3, pp, art, nn#1
2a	The caterpillar metamorphosed into a elegant butterfly.
	art, nn#1, vb#2, pp, art, adj#3, nn#1
2b	The caterpillar changed into a beautiful butterfly.
	art, nn#1, vb#1, pp, art, adj#2, nn#1
3a	Fish swim in water.
	nn#1, vb#1, pp, nn#1
3b	Birds fly in the air.
	nn#1, vb#1, pp, art, nn#1
4a	They believed the red bus was environmentally friendly.
	pn, vb#2, art, adj#1, nn#1, vb#1, adv#1, adj#2
4b	They put their faith in the train being green.
	pn, vb#1, pn, nn#2, pp, art, n#1, v#1, adj#2
5a	To drive a manual car, you must press down the clutch.
	to, vb#1, art, adj#2, nn#1, pn, aux, vb#1, pp, art, nn#6
5b	To open the window, the mouse has to be double clicked.
	to, vb#11, art, nn#8, art, nn#4, aux, to, vb#1, adv#3, vb#1

Table 7.1a: Human tagging for ten pairs data set giving the part of speech and WordNet meaning. (cont...)

Sentence ID	Sentence and type tagging with WordNet meaning
6a	The green grass glimmered as the sun shone on the morning dew.
	art, adj#1, nn#1, vb#1, conj, art, nn#1, vb#2, pp, art, nn#1, nn#1
6b	The ancient building had stood on that small hill for eons.
	art, adj#2, nn#1, aux, vb#3, pp, pn, adj#1, nn#1, pp, nn#1
7a	The Persian cat sat on the carpet.
	art, adj#1, nn#1, vb#3, pp, art, nn#1
7b	The Persian rug was on the dresser.
	art, adj#1, nn#1, vb#1, pp, art, nn#1
8a	The exploded diagram shows how cars work.
	art, adj#2, nn#1, vb#8, conj, nn#1, vb#4
8b	The car exploded at the art show.
	art, nn#1, vb#5, pp, art, nn#2, nn#3
9a	Woman, without her man, is nothing.
	nn#2, pp, pn, nn#6, vb#1, nn#1
9b	Woman: without her, man is nothing.
	nn#2, pp, pn, nn#3, vb#1, nn#1
10a	Trees need sunlight and water to grow.
	nn#1, vb#2, nn#1, conj, nn#6, to, vb#3
10b	Food and drink are essential for your development.
	nn#2, conj, nn#3, vb#1, adj#1, pp, pn, nn#7

Table 7.1b: (...cont) Human tagging for ten pairs data set giving the part of speech and WordNet meaning.

There were minor discrepancies between the two sets of tagging. In most cases, it was where WordNet offered two very similar meanings as in for “sat” where one selected the verb “take a seat” and the other “be seated.”

There were examples where participant 2 chose a valid meaning but missed a clearly closer meaning. In sentence 5b, “Open” was chosen by participant 2 as “cause to open” whereas the meaning selected by participant 1 was “Display the contents of a file or start an application as on a computer”.

A third issue where in sentence 4b, participant 2 selected that none of the meanings were correct for “green” and so did not select a meaning which was contrary to the instructions. Therefore, it was decided to use the results from participant 1 which are those shown in table 7.1.

7.4 Automatic Disambiguation

As discussed in section 2.3.3 association is a fundamental part of deciding which of the possible meanings is the intended meaning for a particular context. While the precise meaning is dependent on the meaning of the sentence as a whole, often considering the likeliness of an event occurring in reality, association can be used to distinguish meanings in the more general case.

When a word is used with a particular meaning, this will normally then be associated with a particular set of ideas. A co-occurrence of associated words, often occurs as their meaning is combined to form a more complex meaning such as a sentence.

The actions that would be associated with colon as part of the anatomy will be different from that of the punctuation mark. So if the "colon is causing you pain", it is more likely that the intended meaning of colon is the former meaning.

When the usage of the meaning of a word is common, other meanings will frequently be used in conjunction with this meaning, to convey events which have happened. By identifying common co-occurrence, it is possible to judge what the intended meaning of a polysemous word is, since the words in conjunction set the context.

In the last chapter, it was shown how the frequency of occurrence of a single word helped identify the contribution that each meaning was making to a sentence. Although the corpus can be used to find the co-occurrence of ideas, this still leaves two issues. The first is in identifying the groups of words which have an association. The second is distinguishing the intended meaning of the word once a grouping has been identified.

Commonly occurring words will have a substantial number of connections to other words. Pairs of closely occurring words can identify collocations such as "red wine" or "sports car". The more commonly occurring pairs of words which appear close to each other can be enough for a person to distinguish the meaning. But a particular meaning of a word can function with many different other words.

This means that the pairs of words alone are not enough to identify all examples of the same meaning for the form of a word. Likewise, even though the most common occurrence of a pair of words might set the meaning for both words, this does not mean that the only possible meaning when the terms co-occur is the same. However, if this is extended to include multiple word co-occurrences then it begins to be possible to collate terms and resolve overlaps.

An example of resolving a meaning could be the meaning of the words "fish" and "chips" which would most dominantly occur in the context of food. However, both words are also terms in poker and if all three words ("fish", "poker" and "chips") were to occur together it might be possible to refine the context.

Similarly, the word "spear" found with "wound" and "blood", could start to lead towards a connection to the meaning weapon.

A clustering of terms can be used to classify sets and build meanings using concepts. This could allow for multiple meanings of the same form to be distinguished purely from the terms with which they occur.

So a triple of "poker - cards - chips" could form a set. This can be used to distinguish the meaning of "poker" from another set "poker - fireplace - fire - toast" based on surrounding keywords. Even when there are no defining terms co-occurring, it could be possible to use overlapping words in a set to form a link. So two sets of "cat - whiskers - fur - animal" could be connected to "beast - animal - monster" to link the use of "whiskers" and "beast" in the same sentence to the meaning of these sets via the word "animal".

The problem is that while a large corpus could be used to build such connections and

information, they still do not align to the ontological lexical database being used as the knowledge source of SARUMAN.

The co-occurrence of terms shows an association, whilst an ontology represents the meaning. So even though both can be used to build a structure that can be used for comparison, linking the two together without human knowledge is challenging.

Pearce et al. (2011) describes how the ontological relationships could be extracted from a human source of knowledge for word meanings. The ontological relationships can be extracted from a definition in order to represent words as a set of properties.

A dictionary definition will often contain ontological links to properties and examples to usage, which could form associations with commonly co-occurring words. By reducing a definition to its keywords a structure similar to set of associated terms is formed. While this will not have many of the associations that can be built from a corpus because it is a small set, the properties can also be used to form a connection between non overlapping terms.

Take examples "bird" and "bat"; both animals might both have the word "wings" in their properties to disambiguate the meaning. If the definition of "bird" as an animal contained within the definition "has wings" and a "bat" had a definition with "is a winged mammal", then the meaning of both words would have added a property of "wing" as part of their sets of properties. In which case, this link ("wing") between the sets of properties, could be acting as an association between these meanings for "bird" and "bat" and have been used to disambiguate the meaning to the meanings containing "wing".

7.4.1 Using WordNet's Definitions

WordNet's definitions have been used for disambiguating meanings by Banerjee and Pedersen (2003), and a similar idea is used here for the sentence level. WordNet contains a definition for almost every word and where there is not one, the first connecting node can be used to build a set comprising the key words of the connected node's definitions.

Each definition is crudely reduced to a group of keywords / lemmas by using a stemmer on each word in the sentence, plus the original word, and if the resultant stem word occurs in WordNet (with an order of precedence of noun, verb, adjective, adverb) then a keyword is added to a set.

A slight weakness in this approach results that not all "stop words" were necessarily excluded that a more sophisticated corpus method would extract. This means that there risks being some extraneous words. Additionally, the structure of the definitions and depth is highly variable within WordNet

Another limit of this approach as was the case in Pearce et al. (2011), is that the definition still often contains words which are polysemous and so have ambiguity. If two words are compared with a common form as the keyword, it could well be the case that this will then give an overlap when they should have been keywords distinct from one another. So "computer" could contain a link to the word "window" as to could the word "porthole" link to "window", then there would potentially be an unwanted overlap.

7.4.2 Disambiguation Weighting Module

In some cases an exclusion based upon the part of speech would already allow several possible meanings with different parts of speech to be excluded. If the word must be a noun, then meanings that would be a verb are not considered by part of the disambiguation.

The first stage of the automatic disambiguation is to give each meaning a weight. This is not the final disambiguation weight, but an intermediate step to determining so.

When comparing two meanings, each meaning is substituted with its set of lemmas and the count of the number of the overlapping lemmas are returned.

Then the square of the maximum overlap, between a meaning and all the meanings of another word, is used as a contribution to the weight. The weight is added to the meaning

being compared and evenly distributed between the meanings for the word which gives the maximum overlap.

If there were three meanings in the word which overlapped, with two of the lemmas representing the single meaning then there would be a weight of 1.333 added to each of the three meanings weights and 4 to the weight of the meaning.

This process is repeated for every word and for every single possible meaning other than the one the meaning which is being compared. The steps to find the meaning are:

For every pair of words in the sentence:

1) Find the 'N' pairs of meanings between two words with the maximum keyword overlap.

For the 'N' pairs :

2) Add to the meaning weight for both meanings, the square of the overlap, divided by 'N'.

Finally, the maximum meaning weight for each word is given a disambiguation weight of 1 and the others 0.

An example, using three words: Word A; Word B; and Word C is given below and illustrated in the figures 7.1-7.4.

Word A has 4 possible meanings (A1, A2, A3 and A4).

Word B has 3 possible meanings (B1, B2, B3 and B4).

Word C has 5 possible meanings (C1, C2, C3, C4 and C5).

Each meaning has had a generated set of keywords which can be compared against the keywords of another meaning. Each comparison will return the count of the number of overlapping keywords.

So the first step is to compare Word A and Word B and then find the meanings with the highest overlap.

Word A	Word B	Word C
A1	B1	C1
A2	B2	C2
A3	B3	C3
A4		C4
		C5

Figure 7.1: An illustration of the meanings for the three word example sentence (word A, Word B & Word C)

In this example, there is one pair of meanings (A3-B3) with a higher overlap than all the others. The number of keywords is 2. (So following step 2) there is a weight of 4 to be added to each meaning of the pair (A3 and B3) as show in figure 7.2.

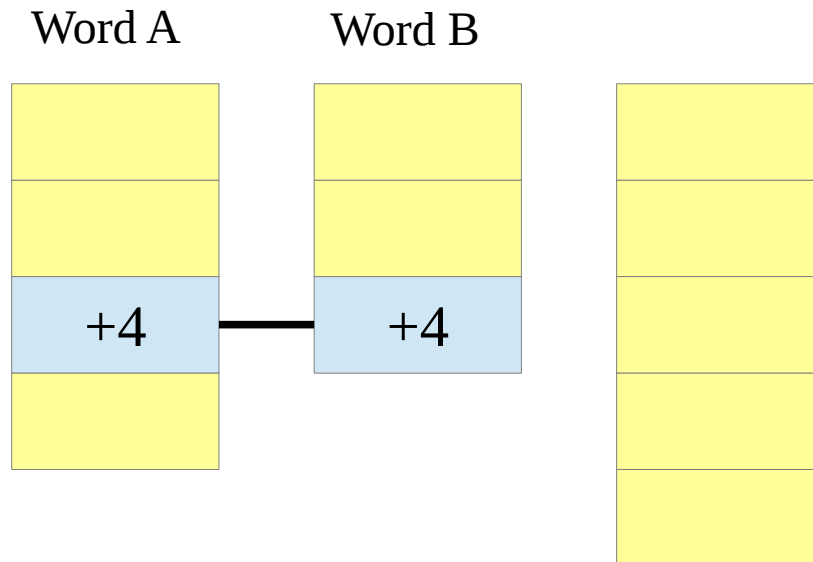


Figure 7.2: Diagram showing the weights between words A & B with a maximum 2 keyword overlap between meanings A3 & B3.

The next comparison is between word A and word C. On this occasion, again the maximum overlap is taken as 2 keywords but this time, two pairs share the same overlap

(A4-C4 and A4-C5). Now the weight is 2 as there are 2 pairs. However, weight A4 has this weight added twice for a total increase of +4 (Figure 7.3).

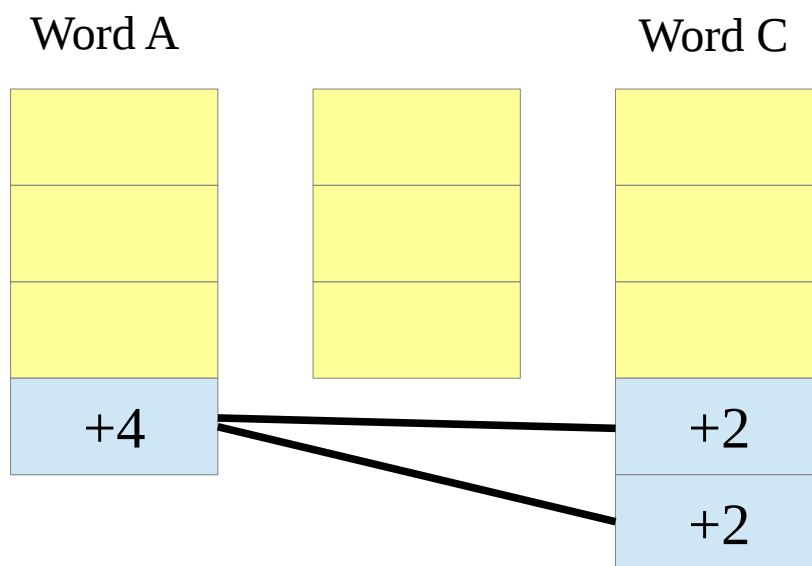


Figure 7.3: Weights for words A & C again with a maximum overlap of 2 but split between pairs A4 & C4 and A4 & C5.

In this example, there is only one more combination of words to make, which is the pairing between Word B and Word C. On this occasion the maximum overlap is taken as 3 keywords with 3 pairs all having an overlap of 3 keywords (B1-C1, B1-C2 and B3-C4).

The total weight is 3 (9 / 3) and so B1 has 6 added to its weight whereas B3, C1, C2 and C4 all have 3 added. This then leads to the situation shown in figure 7.4. The shaded meanings in Figure 7.4 represent the highest score for each word and selected meaning. However, word A has two possible meanings with equal weight.

The maximum weight for word B is B3 and the maximum for word C is C4. So both of these meanings would be selected for the word. Which is done by setting the other disambiguation weights to 0 (i.e. C1, C2, C3, C5, B1 and B2), and C4 and B3 both to 1.

With Word A it is slightly more complicated because there are two meanings with the same weight. In this instance both meanings (A3 and A4) remain valid for further consideration via setting their disambiguation weight to 1 whereas A1 and A2 are set to 0.

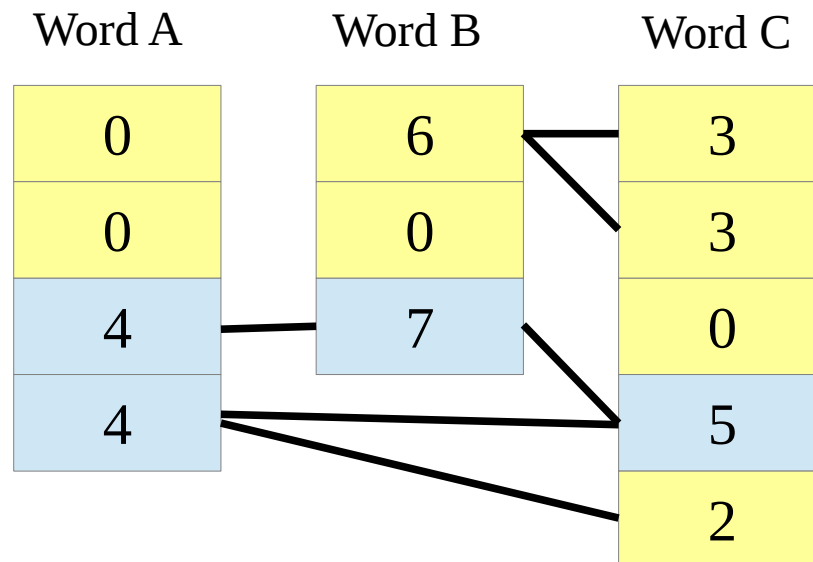


Figure 7.4: The final result of the meaning weights shows lines between all of the maximum connections.

Even though more sophisticated automatic disambiguation could also be considered to compare the words as definitions as opposed to sets of words, this becomes cyclical and could effectively introduce another sentence similarity model as part of the metric. Were LSA used for example to refine this comparison then since LSA is already outperforming SARUMAN, this would invalidate the experimental method of isolating components for SARUMAN.

7.5 Experiments

The experimental benchmark for the numerical version of SARUMAN is used as the starting point for the experiments presented in section 7.6. 4 sets of experimental results for the ten pairs dataset are examined. First, SARUMAN is run as before but uses the tagging from table 7.1 to filter the possible meanings by setting the disambiguation weights of unwanted meanings to 0. First by type and then to the specific chosen meaning.

Next the automatic disambiguation outlined in section 7.4 is used once considering all possible meanings and once first restricting the meanings considered to just the valid parts

of speech based on the human tagging.

7.6 Results and Discussion

Table 7.2 gives the numerical results for the 4 disambiguation experiments alongside the benchmarked version of SARUMAN from the last chapter (section 6.7). There is basic disambiguation using the part of speech as a filter labelled “human tagged type”. The other three experiments aim to select a single meaning. The idealised version of the tagging using the human tagging from table 7.1, labelled “human tagged meaning” and the two automatic disambiguation using the method described in section 7.4.3, one with filtering first by type.

Disambiguation reduces the similarity scores from the mathematical model which was taking the highest possible meaning comparison for each word. In some cases a dramatic reduction from SARUMAN's score can result from the human meanings, but in several cases disambiguation gives a minimal reduction.

ID	Human	SARUMAN raw	Human tagged meaning	Human tagged type	Automatic Disamb.	Type + Auto. Disamb.
1	0.78	0.667	0.597	0.577	0.468	0.475
2	0.91	0.833	0.816	0.833	0.654	0.778
3	0.27	0.637	0.580	0.661	0.114	0.106
4	0.44	0.544	0.370	0.403	0.142	0.168
5	0.24	0.517	0.146	0.284	0.220	0.142
6	0.07	0.440	0.284	0.363	0.184	0.117
7	0.18	0.792	0.635	0.510	0.750	0.401
8	0.11	0.753	0.208	0.431	0.539	0.188
9	0.37	1.000	0.760	0.998	1.000	0.996
10	0.37	0.483	0.151	0.351	0.151	0.148

Table 7.2: Numerical values for SARUMAN with disambiguation

SARUMAN version	PCF	Spearman's	RMS
Raw	0.283	0.334	0.397
Human tagged meaning	0.567	0.486	0.248
Human tagged type	0.476	0.432	0.300
Automatic disambiguation	0.210	0.024	0.352
Auto. Disamb. + type	0.547	0.517	0.268

Table 7.3: The correlations for SARUMAN with disambiguation

Table 7.3 shows the summary metrics of the performance for each of the four disambiguations, alongside the baseline of the original SARUMAN model described in chapter 6. Figures 7.5-7.9 show the graphical representation and it can be clearly seen how, with the exception of automatic disambiguation, that the output is much closer to the human rating with disambiguation.

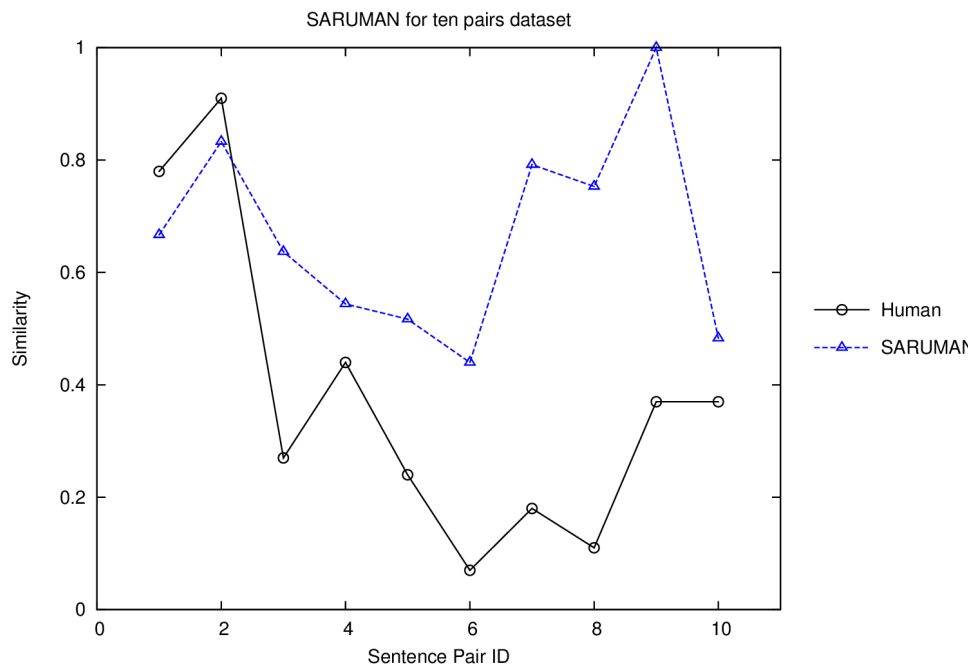


Figure 7.5: SARUMAN with no disambiguation

Figure 7.5 gives the graphical representation of SARUMAN (Chapter 6) on the ten pairs dataset. SARUMAN does show some resemblance to the relative positions of the peaks and troughs but nonetheless, shows a poor match to the human values with a significant overestimate, apart from those scored as highly similar by the human raters.

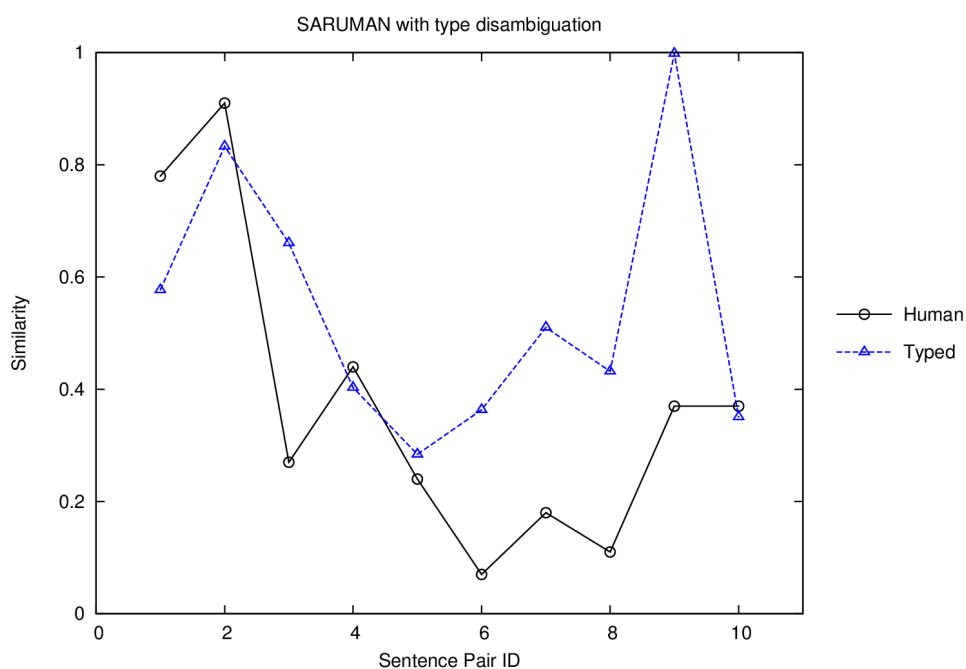


Figure 7.6: SARUMAN with disambiguation of type

Disambiguation of type (figure 7.6) has moved the values much closer to the human scores and there is a clear visual improvement in shape as compared to that from figure 7.1.

The introduction of human disambiguation of meanings (figure 7.7) gives a value much more similar to the human scores than the previous model and even successfully reduces the calculation pair 9 below unity. It is also the case that with human meanings the results are still distinctly different from the human ratings.

Automatic disambiguation (Figure 7.8) has allowed a slight improvement on the values on the right hand side of the graph, but has started to underestimate the scores at the left hand end of the graph. With some improvement over SARUMAN alone it also shows many failings to correspond to the human values and markedly different to the human tagged values of meaning in figure 7.7.

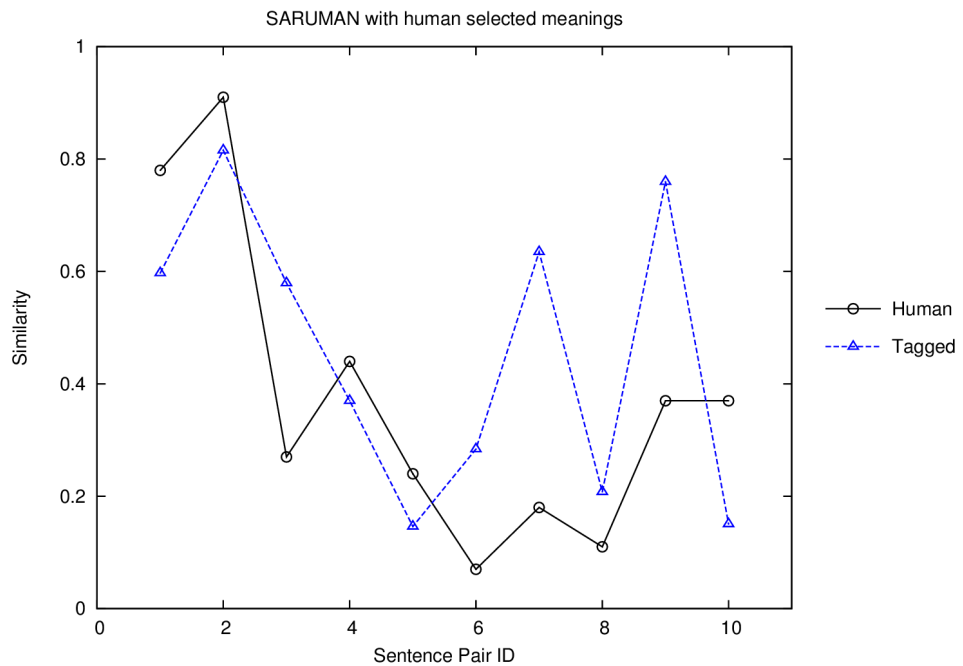


Figure 7.7: SARUMAN with human tagged disambiguation

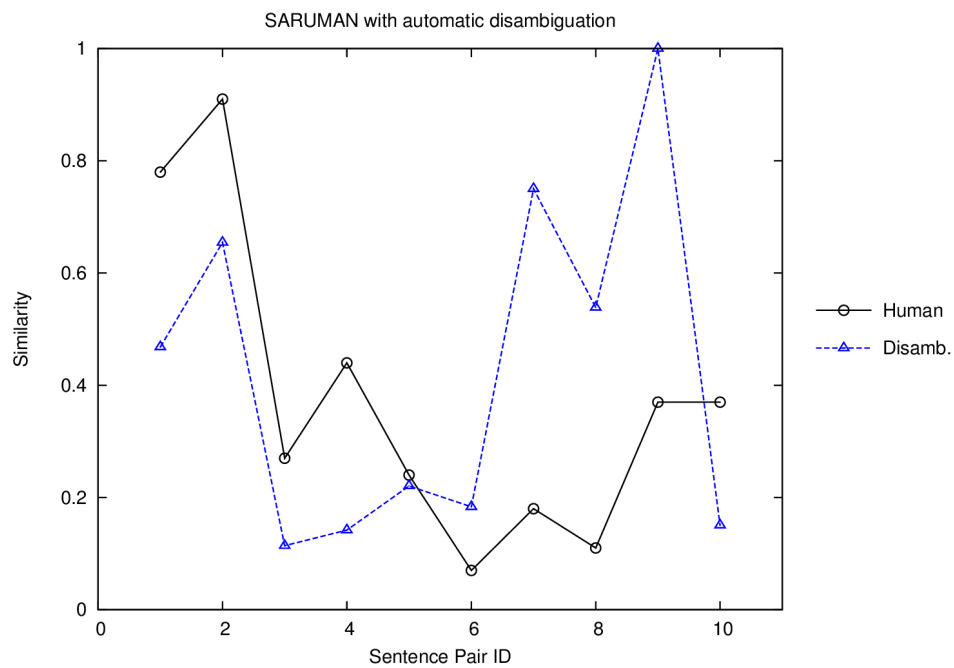


Figure 7.8: SARUMAN with automatic disambiguation

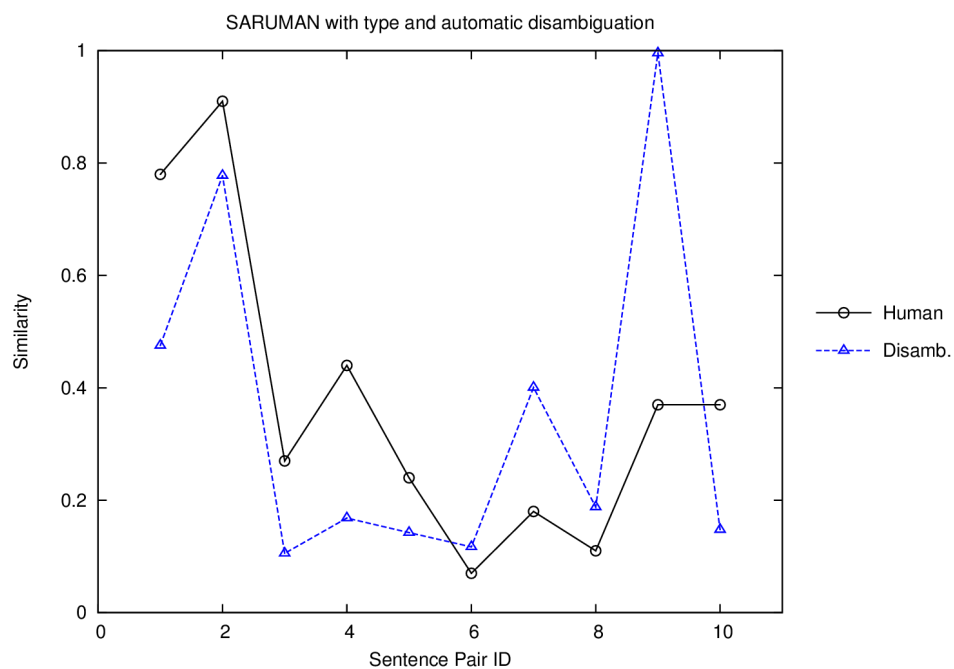


Figure 7.9: SARUMAN with automatic disambiguation of meaning with tagged type

The filtering of type before automatic disambiguation (figure 7.9) gives a clear improvement over disambiguation alone. It still fails to distinguish pair 9 from a value of unity. The shape is clearly closer to the improved values and shows a slight resemblance to characteristics disambiguation by type and human disambiguation (figure 7.7).

7.6.1 Human Meanings and POS Tagging

A clear improvement in performance arises from restricting the comparison to only the valid parts of speech, with a dramatic jump in Pearson's correlation of over 0.19 between the mathematical version of SARUMAN and when it is extended to include disambiguation by type.

The human tagged meanings also show statistically significant improvement over simple disambiguation by type. This shows it is clearly possible to obtain an improvement in performance by including a human level of disambiguation of meaning. The result also represents a limit of disambiguation with the current SARUMAN model, since a machine cannot realistically improve human disambiguation.

An issue arises where on occasion selecting the right meaning deteriorates the performance, not simply in correlation, but in the individual similarity comparison. This is because of the structure of the ontology within WordNet. Some of the structure is very sparse so although a better choice of meaning is made, the structure of the wrong meaning for the context might have been closer.

A more refined meaning might be possible using a properties model and would likely show a clearer benefit even with an approximate method. The reason being here is lost information when using the hypernym relation alone, which can be stored with a properties representation. The different meanings can share a parent node and there is no further detail to refine the comparison. Nonetheless, there were many cases where selecting the correct meaning demonstrated a clear improvement in correlation.

Finally, even when the wrong meaning is found, the overlapping spelling of the meanings as a single word has arisen from an historical link. Which means that even the wrong pair of meanings could be a reasonable approximation of the meaning structure. So while disambiguation improves the model it is less critical to the similarity.

7.6.2 Automatic Disambiguation

There are a number of instances where automatic disambiguation is correctly selecting the correct definitions such as the triple of mouse - open - window. With "open" having 29 definitions, "window" 8 definitions and "mouse" 6 possible definitions.

So selecting the verb "open" definition in WordNet of:

"display the contents of a file or start an application as on a computer"

The noun "mouse" has a definition:

"a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the device is a ball that rolls on the surface of the pad; 'a mouse takes much more room than a trackball'"

and the noun "window" has a definition:

"(computer science) a rectangular part of a computer screen that contains a display different from the rest of the screen)"

The keywords of computer, screen and display all causing a match.

However, there were many cases where the right meaning is not being found. For example, the word "bus" in sentence 4a is selected as the following:

"an electrical conductor that makes a common connection between several circuits; the busbar in this computer can transmit data either way between any two components of the system"

There were almost no keyword matches for any keywords of the definition of bus and the overlap with the third meaning of the adjective "friendly" :

"easy to understand or use; user-friendly computers; a consumer-friendly policy; a reader-friendly novel"

The two groups of keywords had both contained the word "computer" which was enough for the meaning of bus to be selected from its keywords.

The structure of the definitions, inclusion of example sentences for usage and general terms such as verbs or the word "common" above, are often only loosely linked to the meaning in the example. However, the biggest issue to the performance of the automatic disambiguation is the large number of meanings for some words, often with very close meanings in the WordNet definitions.

The raw automatic disambiguation was failing to improve the correlation. Although a slight improvement in the RMS occurs, this process is actually worsening the model. Because not every correct meaning is leading to an improved comparison, the partial success in identifying correct meanings is not leading to improvement in the model.

In some cases, the right meaning in one sentence will compare weakly with the wrong meaning in the other sentence, which then masks the context of comparison. In others, the difference in structure in WordNet's ontology leads to a minimal change in similarity. Therefore, the wrong meanings are dominating the poor performance.

In contrast, when first restricting the meanings to only the valid parts of speech, leads to a starkly contrasting improvement in performance. This is not simply because the automatic disambiguation was choosing meanings which were for the wrong type of speech but due to the influence to the weights from the other meanings that are now excluded.

When the comparison space is reduced, the selection of the correct meaning improved. However, the selected meanings were still also often different from the human selected meaning.

The result of the correlation suggests that now the automatic disambiguation was performing at the same level or slightly above human disambiguation, but because the results are using the wrong meanings at time, this is clearly not the case.

When compared against the performance of the human disambiguation, the automatic version with type tagging is still only giving a PCF of 0.8 compared to 0.89 when compared to without type tagging.

In essence the two versions of SARUMAN with differing disambiguation tagged and automatic, are not directly comparable in the way that the incremental experiment was designed. Both are showing statistically significant improvement over tagging but there is strong sensitivity to the word similarity and the relative score is partially an artefact of the representation of the meaning.

7.7 Conclusions

It has successfully been shown that the idea of including disambiguation to a sentence similarity model has the potential to improve the model's performance via the use of the tagged dataset. This shows the importance of context and disambiguation to meaning and that the inclusion of the Linguistic concept giving an improved performance over the mathematical version of SARUMAN.

The approach for making automatic disambiguation decisions is effective in many instances using the keyword sets built from dictionary definitions within WordNet. This, when limited with the correct part of speech for each word, gave statistically significant improvement over simply tagging the part of speech.

The automatic disambiguation is a way from matching a human level of disambiguation

and when allowing many unwanted definitions, can deteriorate the performance of sentence similarity.

If this were to be the final version of SARUMAN then this approach for automatic disambiguation alongside type disambiguation should be included. However, further Linguistic concepts are still to be added to the sentence similarity model and to exclude the right meanings could mask later improvements.

The meaning disambiguation shows some promise but is very computationally expensive in its current implementation, with thousands of sets of words to be compared for each sentence. Currently the implementation is outside of real-time (taking longer to execute than the creation of a sentence) and would require a high level of optimisation to accomplish real-time.

These two issues in combination means that the version of SARUMAN with type tagging will be used for adding further Linguistic concepts but without automatic meaning disambiguation.

The human tagging will still be used as an indicator for the upper limit of what could be achieved with automatic disambiguation.

While the automatic disambiguation showed itself susceptible to finding wrong meanings, the inclusion was a significant contribution to sentence similarity. It showed that a partial success in finding the right meanings could improve the performance of the sentence similarity model. What was definitely shown was that excluding inappropriate meanings from other parts of speech can improve a mathematical sentence similarity model. This shows that Linguistic concepts can be important to sentence similarity which is meeting the objective of this research (section 1.4). There are, however, many more Linguistic components to be added to SARUMAN and this continues in the next chapter.

8.0 Disambiguation of Clauses

8.1 Introduction

The last chapter showed how the inclusion of disambiguation could improve the performance of the mathematical version of SARUMAN. This was demonstrating that the Linguistic concept of context as outlined in section 2.3.3, was indeed influential to a sentence similarity model.

This chapter continues with the core experimentation outlined in chapter 4 with the purpose of pursuing the objective to show how key Linguistic components can be introduced to a sentence similarity model in order to improve its performance. This chapter focuses on the function of clauses and adds basic word interaction to SARUMAN.

The Linguistic approach being followed in this thesis means that this is the first time that sentences have been specifically divided into the Linguistic functional using more than individual words to compare the semantic similarity of the sentence. As part of accomplishing this other novel steps were included; an extension of the vocabulary, how to compare words of different types and a sequential parser.

Section 2.4 gave details about how words can interact as clauses to form more complex meanings and word interaction as discussed in section 2.3.2, is a fundamental Linguistic component that has been included in the framework (detailed in chapter 4).

The previous chapter was an investigation into the effect of selecting an intended meaning for each word for the sentence similarity model. The version of SARUMAN with disambiguation by type is used for the base to which clause information will be added. Full disambiguation of meaning had been included as an experiment as discussed in the previous chapter's conclusions, this is not well suited for continuing experimentation as it was noisy.

The disambiguation by type is automated by a parser module in SARUMAN and added to

this clausal information. With the previous model, the words were still being freely compared to any other word in the other sentence, irrespective of how the words are functioning together to form more complex units, which from a Linguistics perspective is undesirable.

This chapter will introduce clause disambiguation and a parser module to further remove inappropriate comparisons. While a parser module was needed in order to add the part of speech tagging used for disambiguation of type, much greater emphasis is placed on the parser when including clausal information. This chapter starts by presenting the architecture of the novel sequential parser created for use with this latest version of SARUMAN.

In addition to the parser and clauses, the idea of cross-type comparison is introduced. So that words with a different part of speech can be compared against one another. This is an important change as Linguistically there is a potential semantic overlap between groups with a different grammatical classification. This potentially changes the previous benchmarks so the core results without the automatic parser are included prior to the inclusion of the clausal information.

8.2 Parser

This chapter introduces an innovative deterministic parser based upon Linguistic rules to identify the patterns in English. It is based upon the same vocabulary as stored in SARUMAN's knowledge base (WordNet's vocabulary with the other parts of speech added) and specifically adds the tags that are wanted for SARUMAN to compare the sentences. In part because of the non-standard tags, the parser has not been evaluated separately but a more general parser could also have been adapted to give the same tags. The use of the same vocabulary as was used for the rest of knowledge base reduces discrepancies where the parser returns a part of speech that has been omitted by WordNet.

While another parser could equally well be adapted to fulfil the function of the parser module, a primary concern was in ensuring that the processing remained in real-time,

which would not necessarily have been the case if using an API.

The information required to match the same tagging output as was used by the sequential parser would be present within the calculation of another parser, but it would be duplicating the calculation were this to be done from the output of another model.

8.2.1 Parser Module Objective

The aim of the parser module is not to produce a stand alone parser to give the standard Linguistic rules, but to have a parser which provides the parts of speech and clause tags that will be needed by SARUMAN as it is developed in the following chapters.

As a result, there is more of an interest in the functional structure than in the Linguistic classification which the standard parsers use for their output of clauses such as the Penn treebank (Marcus et al., 1993). However, the same underlying Linguistic rules resolve both situations, and in essence, the same problem is being solved by a parser, just with slightly different resolution in the output.

The parser has to produce the parts of speech for each word and to divide a sentence into its basic noun, prepositional and verb clauses which could then be used later to form more complex clauses.

An additional focus for the parser module, as opposed to the usual academic interest, was speed of execution so as to ensure that the sentence similarity model was capable of real-time operation.

It would have been possible to adapt an existing parser, with access to its source code, to give the desired output, but the differing output and interest in run-time led to a new sequential parser being built.

Section 8.1.3 gives a very simple set of patterns with an example to show why the

sequential parser has advantages.

8.2.2 Differences to General Purpose Parsers

The information that provides the accuracy for the parser comes from the Linguistic rules and patterns. Even with total knowledge of all the possible patterns, there remains situations in English which can have perfect ambiguity (Section 2.4) with more than one valid parsing of a section of a sentence. A human will normally resolve these cases of perfect ambiguity based upon the underlying meaning of the sentence.

Unlike the issues of disambiguation of meaning and the issues involved in managing to automate this process discussed in the last chapter, most sentences can be successfully resolved to a single valid set of patterns and parsing. This is reflected by the very high accuracies achieved by the automatic parsers (Manning, 2011). Even in these cases, there can be a large number of local combinations that have to be handled, but the structural words will often resolve any ambiguity (Quirk, 1962).

The parser is coupled with the sentence similarity model and there is a greater emphasis on the Linguistic function of the words than on the general Linguistic classification. It uses same vocabulary for both meaning and parser (primarily WordNet plus other parts of speech – see section 8.3). This leads to a few cases where greater resolution is required than from the standard treebank classification (Marcus et al., 1993), because the present participle of the verb which can occur in many different contexts. The clauses being found are the basic noun clauses, verb clauses and prepositional clauses identified as to whether they function as subject, object or subordinate clause (section 8.2.7).

There is a push-pull between being comprehensive, and being accurate. When there are common words which have obscure meanings with a different part of speech to the common meaning, then an automatic parser may have to consider possible combinations for the clauses and parts of speech that a human can ignore from the context. While a parser should still find the valid pattern, this could be coupled with another valid pattern

which is unlikely from the meanings. Therefore, there is a further disambiguation to be resolved which usually should have been resolved in favour of the commoner meanings. So statistically, most of the time only using the common words was already giving the desired disambiguation , but with the extra choice the automatic parser may reduce the number of times that the correct choice is made.

Likewise being aware of obscure patterns can result in finding more situations of perfect ambiguity which were previously resulting in the most likely to occur disambiguation, prior to their inclusion. Despite this there are many situations where the extra rules or meanings make no difference to the final resolution of the parsing, but only require that more patterns have to be considered before deciding upon the same pattern as before their inclusion.

There are, therefore, two factors which contribute to the accuracy of a parser. These are:

1. Identifying the possible patterns .
2. Disambiguating the situations of perfect ambiguity.

By only using WordNet's vocabulary for determining the nouns, verbs, adjectives and adverbs; there are situations where familiar knowable words can be encountered and the parser will not necessarily be able to accurately guess the intended part of speech of the word and hence can struggle with identification. As well as WordNet having many obscure meanings for the parser to contend with, there are some situations where the form of the word will be included but that the usage as a particular part of speech is lacking from the WordNet definitions.

The other way in which the parser could likely perform below the strongest automatic parsers is in resolving perfect ambiguity. When perfect ambiguity is found, the SARUMAN parser chooses the parsing that favours the richer meaning structure in WordNet. This means favouring the nouns followed by verbs. The reason for this is that there is normally semantic overlap between meanings which share a common form and so a more meaningful comparison would be possible, capturing some of the correct information than if an error was to go in the other direction.

The specialist output is shallower than a general Linguistic parser would aim to classify clauses. While the information for both cases should be contained within the calculations of both SARUMAN's sequential parser and the general purpose parsers, to extract and use this information can require some significant program changes. The restrictions on including all of the WordNet meanings and not distinguishing the most likely occurrence, also means that SARUMAN's parser would benefit from being modified to distinguish between the common and rare forms.

The most significant contribution is the inclusion of the output of the parser as part of the sentence similarity model itself, not the accuracy of the parser. The core contribution of the parser itself is its ability to process the complicated rules quickly, not in the rules used themselves. The timings for the other parsers are not included in the literature and some early concerns over processing speed mentioned, are likely of less relevance because of increase in computer power.

The objective to the research is simply that the parser executed in real-time, it is only for the interest of other potential applications using a parser where greater speed might become critical. The description of the sequential parser is therefore focused upon the key aspect of how the parser can run efficiently, rather than attempting to enumerate how to identify the very large number of Linguistic patterns that can exist when considering the ideas of Linguistics briefly covered in section 2.4. The implemented parser managed significantly better than real-time (details of the timings are only presented in chapter 11).

These factors mean that SARUMAN's parser has not been directly compared to any other model and it would require significant modification to make it perform as a general parser which is beyond the scope of this research into sentence similarity.

It is, however, mainly the rules of Linguistics that dictate the potential accuracy of a parser and it would be possible to use the same core principles for building a sequential parser that used a different set of patterns. Probabilities could be added as an additional step once perfect ambiguity has been identified.

The indications are that the SARUMAN parser is working strongly and most of the time

giving the correct parsing for the sentence, as expected. There are occasions where the ambiguity is being resolved differently from the intended meanings (section 8.5) and that some issues arise with words outside of its vocabulary briefly discussed in chapter 13.

The parser copes with many situations that are not really relevant to the assessment of the performance of SARUMAN because the noise from the parser is small, in comparison to the other sources from the representation of the meanings of the words and their disambiguation (as covered in the previous chapter). If the parser were not performing well then it would have been necessary to consider possible error handling and for areas where the parser identified issues, the raw SARUMAN algorithm would have to be used instead.

8.2.3 Simplified Sequential Parser Example

The reason for adopting a sequential parser is that it allows for any identified Linguistic pattern to be used, but still enables fast processing by never making a pattern comparison at run-time. Instead, it is the case that a decision simply based upon one word each time can be used to determine the next state and current action to take in order to parse the sentence.

This section includes an example of why sequential parsers work like this, using a greatly simplified set of patterns. This avoids dealing with the situation of English as whole while the principles remain the same for the sequential parser, regardless of the number of patterns.

A simplified version of the patterns and parts of speech that have to be handled by a parser help to demonstrate how a sequential parser efficiently manages the patterns. The objective is to convert the word of a sentence into a continuous sequence of valid patterns using their parts of speech.

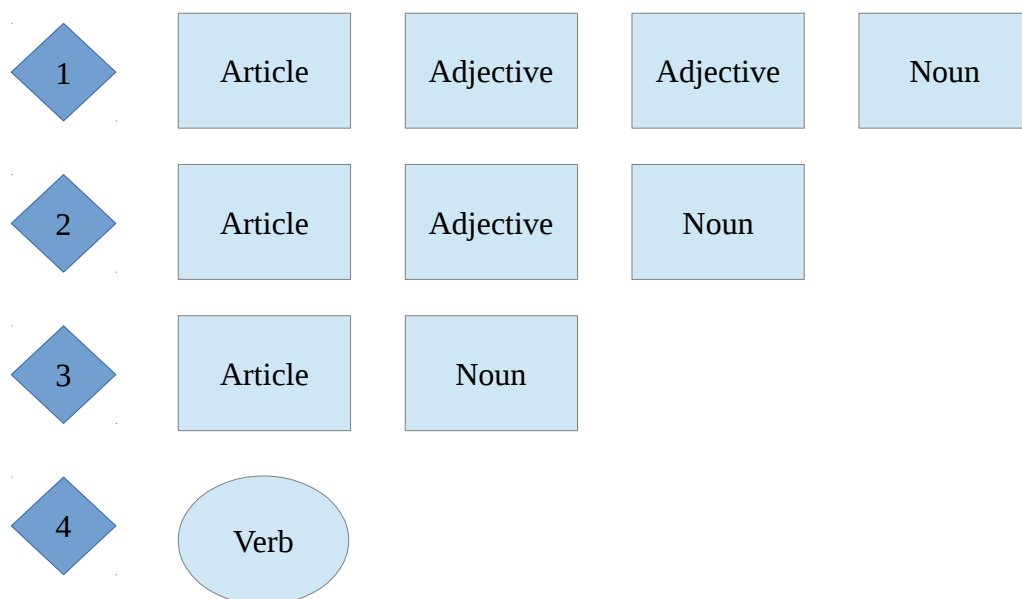


Figure 8.1: The four patterns considered valid for the simple example for sequential parser.

Figure 8.1 shows 4 basic patterns. For the sake of further simplicity, only four parts of speech will be used with: article = A; adjective = J; noun = N; and verb = V.

The 4 complete valid patterns become:

- <1> = A-J-J-N
- <2> = A-J-N
- <3> = A-N
- <4> = V

A completed pattern represents a closed clause and can then be followed by another pattern. To process the sentence a word at a time, it also needs to consider the sequences that could be going on to form a complete pattern with the addition of the subsequent words, not just patterns that are already complete.

So for this example, there are 9 possible sequences to consider, as shown in table 8.1.

There are 3 open clauses which could be forming parts of a complete clause (A, A-J, A-J-J). There are two possible sequences that would mean that there is not a valid description of the entire input, so a sequence starting with an N or a J would be an error in the grammar.

Sequence	Clause state	Pattern
A	Open	
A-J	Open	
A-J-J	Open	
J	Invalid closed	
N	Invalid closed	
V	Closed	<4>
A-N	Closed	<3>
A-J-N	Closed	<2>
A-J-J-N	Closed	<1>

Table 8.1: The valid possible sequences that could form valid clauses for the example patterns in figure 8.1.

When every input only has one part of speech then there is no disambiguation to consider. You would simply need to wait for the pattern to end in order to describe the whole sentence in terms of the patterns.

So the next stage is to consider a situation where one of the inputs has more than one possible part of speech with an adjective or a noun, symbolised by J/N.

Taking the input sequence as A, J/N then V, there are two approaches to consider the first is finding all of the valid patterns at each step. When the first word is processed we can eliminate the pattern <4> but the other three patterns are still valid.

	<1>	<2>	<3>	<4>
A + J/N	Yes	Yes	Yes	No

Table 8.2: The result of the valid pattern comparison for an article followed by a word which could be either an adjective or noun.

The same is true when we add in the second word as shown in table 8.2. With the third word there are two possible sequences to consider A-J-N and A-N-V. These patterns would then have to be compared against the three remaining patterns to find the valid patterns.

Combination	<1>	<2>	<3>	<4>
A-J-V	No	No	No	-
A-N-V	No	No	Yes	-

Table 8.3: The continuation of the possible pattern test from table 8.2 with an added verb to give to combinations to consider.

Pattern <4> could already be excluded and only one valid pattern remains. Table 8.3 shows the remaining matches that would have to be formed and there is only one valid pattern which shows that the correct parsing was <3> + <4> with A-N + V. However, this required a total of 10 comparisons to process the sequence to find the valid parsing.

As there are only a finite number of parts of speech, it would be possible to avoid the direct comparisons and make a judgement based upon the current sequence. For A-J and A-N the patterns that would result for each part of speech added next are given in tables 8.4 & 8.5.

Sequence	+	Valid	States
A-J	A	No	[A-J] + A
	J	Yes	A-J-J
	N	Yes	<3>
	V	No	[A-J] + <4>

Table 8.4: The outcome of adding each of the parts of speech to an article adjective.

Sequence	+	Valid	States
A-N	A	Yes	<3> + A
	J	No	<3> + [J]
	N	No	<3> + [N]
	V	Yes	<3> + <4>

Table 8.5: The outcome of adding each of the parts of speech to an article noun.

While some of these sequences would not be possible, by finding the combinations for A-J/N then there would only be two outcomes to consider without the need for further comparisons. As seen in table 8.6, there would then be two combinations (A-J-V, A-N-V), one of which is valid (A-N-V). Compared with the pattern checking in table 5.3, there are 4 fewer comparisons.

Current	+	Combination	Valid	States
A-J/N	V	A-J-V	No	[A-J] + <4>
		A-N-V	Yes	<3> + <4>

Table 8.6: The result of adding a verb to an article followed by an adjective/noun using the sequences from tables 8.4 and 8.5.

Next comes the crux to building a sequential parser, to avoid having any situation which returns more than one possible outcome. This can be accomplished through adding in a new sequence.

Rather than having to consider each combination that there is, only one sequence that results needs to be considered. This is achieved by adding another valid sequence to table 8.1. So while A-J/N is not part of a larger pattern it can be resolved to one. Table 8.7 shows the how such a state would cope with all the possible parts of speech input. Notice that now an extra possible next part of speech has been added and that is if there were another J/N following on after A-J/N.

Sequence	+	States
A-J/N	A	<3> + A
	J	A-J-J
	V	<3>+<4>
	N	<2>
	J/N	A-J-J/N

Table 8.7: The single state output for adding each possible part of speech when treating article adjective/noun as a single sequence - not two possible combinations.

While if the next word had a single state the patterns would have resolved unambiguously, there becomes the need for yet another ambiguous state to be added of A-J-J/N.

The final step for building a sequential parser is to reduce the number of extra states (A-J/N, A-J-J/N) that are present. This can be reduced through either patterns in the language or by being able to take the same action with multiple actions.

The most obvious pattern to reduce is that of an adjective chain which can be of indefinite length. If you had an article followed by an adjective chain and then add an adjective to it then you still have an article followed by adjective chain. So the pattern remains unchanged.

8.2.4 Limitations of the Parser

As with all parsers each word is assumed to have a single function in a sentence so as to mean that a contiguous set of units can be constructed.

However, there exists the possibility in English where clauses elide together so that words can have more than one meaning in the same sentence. An extreme case allows for a word to even possess more than one distinct part of speech at the same time.

An extension of the idea of a zeugma can be found in the following sentence.

"Finally, the nail was hammered and so was I."

Here, the second clause "and so was I" is a sentence which takes an object but none is provided. While this would be an example of implied words, in this instance the subject is provided by the earlier sentence.

So that the sentence could be expanded as

"Finally the nail was hammered and I was hammered."

Here, though the first use of hammered is as a past participle and the second as an adjective. The result is here that the word "hammered" in the original sentence would require 2 fundamentally different parts of speech, but a parser by design can only give one part of speech.

8.2.5 Implementation

Every form has a type which represents all of its possible parts of speech stored in a simple database. Each word is tagged with one of the possible types given in table 8.8.

The type ID is simply a number that is stored to represent the part of speech of a word but can also refer to combinations which are not strictly speaking part of speech. This is in part why the word type was used in the design of the framework (chapter 5). The word in { } is an example of a word inside of the vocabulary in the knowledge base, that is coupled with that particular type.

Type ID	Functional type	Type	Functional type
1	Noun {car}	23	Noun (plural) / Verb (3 rd) {makes}
2	Verb {protect}	24	Pronoun Possessive {my}
3	Noun/Verb {lapse}	25	Pronoun {you}
4	Adjective {floury}	26	Article {the}
5	Adjective/Noun {animal}	27	Verb (Past Participle -pp) {written}
6	Adjective/Verb {alight}	28	Verb (Past) {wrote}
7	Adjective/Noun/Verb {abstract}	29	Verb (Past + Past Participle -ed) {loved}
8	Adverb {ably}	30	Verb (Present Participle -ing) {writing}
9	Adverb/Noun {Sunday}	31	Conjunction {and}
10	Adverb/Verb {multiply}	32	Subordinator {because}
11	Adverb/Noun/Verb {bang}	33	Interrogative Pronoun {what}
12	Adjective/Adverb {easy}	34	As {as}
13	Adjective/Adverb/Noun {easterly}	35	Modal {should}
14	Adjective/Adverb/Verb {direct}	36	Auxiliary have {has}

Table 8.8a: Possible part of speech ids used by the parser (cont...)

Type ID	Functional type	Type ID	Functional type
15	Adjective/Adverb/Noun/Verb {clean}	37	Auxiliary be {was}
16	Preposition {for}	38	Interjection {ouch}
17	To {to}	39	Possessive {cat's}
18	Not {not}	40	Possessive (plural) {cats'}
19	Noun (Plural) {cars}	41	Divide {-}
20	Noun (Singular + Plural) {deer}	42	Like {like}
21	Noun (Singular + Plural) / Verb {fish}	43	Name {Matthew}
22	Verb (3 rd Person Singular -s) {writes}		

Table 8.8b: (...cont) Possible part of speech ids used by the parser

8.2.6 Parser Implementation

It is possible to build a set of rules for any particular sequence of possible inputs in order to identify the possible basic clauses.

A sequential parser works by processing the type of each word, one at a time as seen in the simple example above. The parser has an internal state and a decision is pre-determined for every possible input state. This can call an action to update internal states or add a set of

tags to the output. Then it sets a new internal state.

Any clauses which are separated by punctuation, such as being in speech marks, are processed separately before being recombined. In the case of speech marks, a substitute of a noun clause is used for processing the remaining sentence.

An action can do any of the following:

- Add the current word to an open clause
- Close the current open clause
- Create a new clause and add the word to it
- Update the internal variables

Because clauses can be divided by other clauses such as an interrogative sentence, it is possible that the current clause can be closed and another clause is opened at the same time.

The internal IDs are fairly small in number as they each represent either a level in a basic clause or an ambiguous state and given in table 8.9. The IDs are the result of the Linguistic patterns used, coupled with the number of different actions needed computationally to resolve the part of speech tags.

There are 25 internal IDs which combine with the 43 possible input states to give 1075 states to call an action, although the number of distinct actions is far smaller than this. Because the output of the action is also dependent upon the internal variables the output is not necessarily the same for the same input type and current state.

Internal ID short name	Typical input to cause state
Start	
Head Word Level	Preposition , as, like
Pronoun level	Pronoun, Interrogative
Article level	Pronoun Possessive, Article
Adjective Level	Possessive, Adjective, Verb (pp)
Adjective/Noun Level	Adjective/Noun
Adverb level	Adverb
Modal Level	Modal
Have level	Auxiliary have
Be Level	Auxiliary be
Have been level	Auxiliary be
Verb Clause level	Verb, verb (-ing), verb (-past)
Verb Preposition level	Preposition
Noun clause level	Noun, noun(plural)
Object level	Noun, noun(plural)
Ditransitive level	Noun, noun(plural)
Extend to level	to
Extend subordinate verb level	Verb (-ing), verb (-pp)
Noun or Verb	
Adjective/Noun + Adverb	
Adjective/Noun + Verb (-ing)	
Adjective/Noun + Verb (-pp)	
Adjective/Noun + Verb (-ed)	
A/N + Noun (plural) / Verb (-s)	
End	

Table 8.9 Possible internal states for the sequential parser

8.2.7 Clause Tagging

Once the whole sentence has been processed, the sentence is expressed in terms of one of 4 possible basic clause types or a conjunction. However, the operation of the clauses with respect to the main clause is still to be set. Because the basic clauses no longer contain any ambiguity (as for any pure ambiguous sentences the parser has picked a possible parsing), it is fairly simple to identify which is the main verb clause. The type of the sentence is stored in the main verb clause: interrogative, conditional, descriptive or imperative.

The tense and mood of each verb clause has already been encapsulated in the verb clause, which means a passive sentence has already been identified. The clauses are tagged in a similar manner to the sequential parsing only without ambiguous types. The clause type IDs are needed to resolve the basic clause structures that could be used to form more complicated clauses to make an entire simple sentence (section 2.3). This functional structural information is used directly for advanced word interaction in chapter 10. Table 8.10 shows the 18 possible clause states that the parser can tag.

PC = prepositional clause

Subject = subject clause

Object = object clause

XC = Participle clause e.g. (“to make“, “eagerly hunting” or “pushed in”)

SC = subordinate subject clause

OC = joined object clause

Sub. = Subordinate

And = joined clause

New Sent= Means that there is a completely new sentence that is starting

ID	Clause	ID	Clause	ID	Clause
1	Subject	7	XC	13	Object SC
2	Subject XC	8	Sub. Verb	14	Object PC
3	Subject SC	9	Main PC	15	And Object
4	Subject PC	10	And Verb	16	Ditransitive OC
5	And Subject	11	Object	17	XC OC
6	Verb	12	Object XC	18	New Sent.

Table 8.10 Clause type IDs and description

8.2.8 Parser Example

The following is a walk-through demonstration of the state flow for input sentence :

"The Ginger cat sleeping to my annoyance recently had sat on the green mat."

STEP 1: possible type (from table 8.8) tagging

the, ginger, cat, sleeping, to, my, annoyance, recently, had, sat, on, the, green, mat

26, 7, 3, 30, 17, 24, 1, 8, 36, 29, 16, 26, 7, 7

art., nn/ad/vb, nn/vb, v-ing, to, pn-pos, nn, adv, have, v-ed, prep, art., nn/adj/vb, nn/adj/vb

STEP 2: Sequential state parsing shown in table 8.11.

STEP 3: Add new type and clause tags from the internal structures:

26, 4, 1, 2, 16, 24, 1, 8, 35, 2, 16, 26, 4, 1

art., adj., noun, verb, prep, pn-pos, noun, adv, modal, verb, prep, art., adj., noun

1, 1, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 5, 5

STEP 4: Add clause type (table 8.10) tagging : 1, 7, 17, 6, 11

Clause 1 - Subject clause

Clause 2 - Participle verb clause

Clause 3 - Subordinate object clause

Clause 4 - Main verb clause

Clause 5 - Main Object clause

Verb states are represented for each clause: -1, 8, -1, 18, -1 (-ve means not a verb clause)

Expressed in friendlier terms gives 5 clauses:

Clause 1 - noun clause {the, ginger, cat} {article, adjective, noun}

Clause 2 - extended verb clause {sleeping, to} {verb, preposition} {present tense}

Clause 3 - noun clause {my, annoyance} {pronoun possessive, noun}

Clause 4 - verb clause {recently, sat, on} {adverb, verb, preposition} {had past tense}

Clause 5 - noun clause {the, green, mat} {article, adjective, noun}

	Internal State level	External type	Action	Explanation
0	start	Article	Start new noun clause add "the"	
1	Article	Adj./Noun/Verb	Extend noun clause add noun set clause state adj./noun	Verb is excluded following article as valid type exists
2	Adjective /noun	Noun/Verb	Set current noun in noun clause to adjective; Close noun clause with new noun	Noun (singular) and verb mismatch so verb excluded so add noun.
3	Noun Clause	Verb (-ing)	Start new verb clause	
4	Extend subordinate verb	To		waiting to determine whether preposition or to + infinitive
5	Extend to	Pronoun Pos.	Treat "to" as preposition so add to verb clause; Start new noun clause	
6	Article	Noun	Add noun and close noun clause	Returns to the last noun clause level (nc level)
7	Noun Clause	Adverb	Start verb clause	
8	Adverb	Auxiliary Have	Advance current verb clause state to main verb and modal state	Ambiguous level which can either be an extended noun clause or main verb
9	Have	Verb (-ed)	Close verb clause with "had + past participle"	
10	Verb Clause	Preposition	Add preposition to last verb clause	
11	Verb preposition	Article	New noun clause	
12	Article	Adj./Noun/ Verb	Extend noun clause add noun set clause state adj./noun	
13	Adjective/noun	Adj./Noun/ Verb	Set current noun in noun clause to adj. Set clause state to adj./noun	
14	Adjective/noun	End of Sentence	Close noun clause as noun	

Table 8.11 Demonstration of flow of internal states (from table 8.9) and actions

8.2.9 Parser Results

The test dataset that is needed for continuing the investigation of the framework and SARUMAN is the ten pairs dataset and the results from the parser module are given in table 8.12. Second row is type-tagging; third output type; and fourth clause types.

1a	<i>the Persian cat sat on the carpet</i>
	the, noun/adj, noun/verb, verb(-ed), prep, the, noun/verb
	the, adj, noun, verb, prep, the, noun
	Subj.{3}, VC {2}, Obj. {2}
1b	<i>the ginger cat sat on the mat</i>
	the, noun/adj/verb, nn/vb, -ed, prep, the, noun/adj/verb
	the, adj, noun, verb, prep, the, noun
	Subj.{3}, VC {2}, Obj.{2}
2a	<i>the caterpillar metamorphosed into a elegant butterfly</i>
	the, nn, verb (-ed), prep, the, adj., nn/vb
	the, noun, verb, prep, the, adj, noun
	Subj.{2}, VC {2}, Obj. {3}
2b	<i>the caterpillar changed into a beautiful butterfly</i>
	the, nn, verb (-ed), prep, the, adj, nn/vb
	the, noun, verb, prep, the, adj, noun
	Subj.{2}, VC {2}, Obj. {3}
3a	<i>fish swim in water</i>
	nn (s+p) / vb, nn/vb, prep, nn/vb
	noun, verb, prep, noun
	Subj.{1}, VC {2}, Obj. {1}
3b	<i>birds fly in the air</i>
	nn (p) / verb (-s), nn/adj/vb, prep, the, nn/adj/vb
	noun, verb, prep, the, noun
	Subj.{1}, VC {2}, Obj. {2}

Table 8.12a: Results of parser for ten pairs dataset (cont...)

4a	<i>they believed the red bus was environmentally friendly</i>
	pn, verb (-ed), the, adj/nn, nn/vb, aux (be) , adv, adj/noun
	pronoun, verb, article, adjective, noun, modal, adverb, noun
	Subj.{1}, VC {1}, Obj. s.Subj.{3}, s.VC{2}, s.Obj.{1}
4b	<i>they put their faith in the train being green</i>
	pn, vb(-pp), pn-pos, nn, prep, the, vb/nn, aux(be), nn/adj/vb
	Pronoun, verb, pn-pos, noun, prep, the, noun, modal, adj.
	Subj.{1}, VC {1}, Obj.{2}, PC{3}, s.VC{1}, s.Obj.{1}
5a	<i>to drive a manual car ~comma you must press down the clutch</i>
	to, nn/vb, the, adj/nn, nn, pn, mod., nn/vb, pp, the, nn/vb
	to, vb, the, adj., noun, pronoun, modal, verb, prep, the, nn
	Part. {2}, s.Obj{3}, Subj{1}, VC{3}, Obj.{2}
5b	<i>to open the window ~comma the mouse has to be double clicked</i>
	to, nn/adj/v, the, nn, the, nn/vb, have, to, be, n/a/r/v, -ed
	to, verb, the, noun, the, noun, mod., mod., mod., adverb, verb
	Part. {2}, s.Obj{2}, Subj{2}, VC{5}
6a	<i>the green grass glimmered as the sun shone on the morning dew</i>
	the, n/a/v, n/v, -ed, as, the, n/v, -pp, prep, the, nn, nn
	the, adj., nn, verb, prep, the, noun, verb, prep, the, nn, nn
	Subj. {3}, VC{1}, Prep. {3} , s.VC{2}, s.Obj {3}
6b	<i>the ancient building had stood on that small hill for eons</i>
	the, a/n, v-ing, have, v-ed, prep, pn, n/a/r, n/v, prep, nn-p
	the, nn, verb, modal, verb, prep, pn, adj, noun, prep, noun
	Subj. {2}, Part. {1}, VC{3}, Obj. {3}, Part. {2}
7a	<i>the Persian cat sat on the carpet</i>
	the, adj/nn, nn/vb, v-ed, prep, the, nn/vb
	the, adj, noun, verb, prep, the, noun
	Subj.{3}, VC {2}, Obj.{2}
7b	<i>the Persian rug was on the dresser</i>
	the, adj/nn, nn, aux-be, prep, the, nn
	article, adj, noun, mod, prep, the, noun
	Subj.{3}, VC {2}, Obj.{2}

Table 8.12b: (...cont) Results of parser for ten pairs data set (cont...)

8a	<i>the exploded diagram shows how cars work</i>
	the, v-ed, nn/vb, nn-p/vb-s, sub, nn-p, nn/vb
	the, adj, noun, verb, pronoun, noun, verb
	Subj.{3}, VC{1}, s.SUBJ{2}, s.VC{1}
8b	<i>the car exploded at the art show</i>
	the, nn, v-ed, prep, the, adj/noun, nn/vb
	the, noun, verb, prep, the, adj, noun
	ubj.{2}, VC {2}, Obj.{3}
9a	<i>woman ~comma without her man ~comma is nothing</i>
	nn, prep, adj/nn, nn/vb, aux-be, pn
	noun, prep, adj, noun, modal, pronoun
	Subj.{1}, Prep.{3}, VC{1}, Obj.{1}
9b	<i>woman without her ~comma man is nothing</i>
	nn, prep, adj/nn, nn/vb, aux-be, pn
	noun, prep, noun, noun, modal, pronoun
	s.Subj.{1}, Prep.{2}, Subj.{1}, VC{1}, Obj.{1}
10a	<i>trees need sunlight and water to grow</i>
	nn(-p)/v(-s), nn/vb, nn, conj, nn/vb, to, vb
	noun, verb, noun, conj, noun, to, verb
	Subj.{1}, VC{1}, Obj.{1}, and Obj.{2}, Part. {2}
10b	<i>food and drink are essential for your development</i>
	nn, conj, nn, aux-be, adj/nn, prep, pn-pos, nn
	noun, conj, noun, modal, noun, prep, pn-pos, nn
	Subj.{1}, and Subj.{1}, VC{1}, Obj.{1}, Prep.{3}

Table 8.12c: (...cont) Results of parser for ten pairs data set

The tags are actually internally stored as numbers but slightly more human friendly tags have been included in table 8.12 so as to allow the final parsing to be seen. The clauses are abbreviated and the number of words in each clause given in {}. It can be noted how that some information, such as auxiliary verbs such as “to be”, is reduced to a simple modal tag. This is because all of the verb information is contained within the verb clause tag and the words ignored.

The results show that the output tagging is very close to the human tagging shown in table 7.1. While there is a discrepancy between "her" being classed as an adjective by the parser, this is because the types chosen meant that there was not the option to add "her" as both a pronoun and a possessive pronoun. Therefore, it was added as an adjective/noun by the type tagger. The verb "be" is tagged as an auxiliary verb by the tagger although tagged as a verb in the human tagging. This change is not significant as it will still be handled the same way without meaning tagging.

There is only one meaningful discrepancy in the part of speech output and that is for: "the ancient building" where a perfect ambiguity existed and the parser selected "ancient" as a noun with "building" as a participle. This selection was made on the basis that the noun and verb structure is richer than the adjective and noun structure.

However, although a valid interpretation of the sentence (6b), it is not the parsing which would have been chosen by a person. On a more complicated dataset to parse more discrepancies would be expected.

8.3 Extending the Vocabulary

The parser needs to be able to identify all parts of speech in order to accurately parse a sentence. At the same time as adding the extra forms for the parser, the vocabulary of the knowledge base was increased in order to cope with the other parts of speech, beyond nouns, verbs and adjuncts.

The same approach as was already described for the adjectives (section 6.2.1) can be used for possessives by treating them as an adjective pointing to the meaning of the noun. So would give a meaning of the format "an123".

Pronouns can be linked to related nouns inside of WordNet so "he" can be viewed as "man" with an appended child node. Possessive pronouns can now be treated identically to possessives. So if "he" gave the meaning representation "n!#!^*?" then "him" becomes "an!#!^*?". Proper nouns are handled like pronouns.

Prepositions are shallower in their semantic meaning than the main word types but still have a distinct semantic meaning. The prepositions are grouped into similarly grouped meanings, given a unique root node and where appropriate linked to noun structure. For example "atop" is given a meaning linking to "upper surface".

Other parts of speech are given their own root node and a predetermined depth with minimal distinction of structure inside of the sets. The other groups are articles, conjunctions (including subordinators) and interjections. Interjections can be negative or positive.

Auxiliary verbs do not have a convenient hypernym structure and are discussed in more depth in the next chapter.

Numbers represent an infinite set unto themselves. While the numbers that occur in language as words are small in quantity, the meaning of any number can be interpreted by a person by understanding the individual meaning of its components such as 34.9847cm. Numbers are not handled separately but as if they were stored as individual words in the ontological database.

The interest of a number, for the task of semantic similarity, is not to identify the exact meaning of an individual number but how it relates to other numbers. To this end, an unknown number is related to a known number already, with a meaning stored in WordNet by adding it as a child node of the structure. The hypernym node is based upon the magnitude of the number related to a numerical word within WordNet, such as a "million" or "100" which has the closest numerical value to the added number.

The parent of each number was determined by the next smallest value encapsulated within the knowledge base meanings and this was implemented by a very short subroutine which made a test for a number once a word was not found by the word look up in the framework as part of the WMS (figure 4.2).

8.4 Cross-type Comparison

After the introduction of disambiguation, it can be important to be able to find the similarity between words with different meanings.

So far the comparison between meanings with different types would always score 0. This is because there can be no common hypernym between any of the meanings as the root node is different.

However, the meanings often have a common structure that has resulted from the ontological connections with the noun clause. As an example, a noun with meaning "n1234" being compared against an adjective "an12356", have an overlapping structure from the nouns with the meaning representation of "n123".

The structure of the meaning is not a simple hypernym relationship and it would have been possible to add the attribute differently for cross-type comparisons. If the adjective attributed was appended to the end of the noun meaning then it would have become: "n12356a". This allows for the overlap to be used as if it were the lowest common hypernym (LCH).

Although, strictly speaking the representation is no longer a hypernym chain model but a properties model, this is conceptually possible with the word meaning similarity formula [6.1] because its input was the common overlap, C12, not the LCH.

Therefore, if a pair of meanings are being cross compared then the root nodes can be cut and appended to the end, in order to manage a common overlap. With an adverb linked to a noun, compared to a verb this would require moving 'ra' to the end for the adverb and likewise a 'v' for the verb meaning.

8.5 Clause Disambiguation Implementation

Once the clauses have been labelled, it is possible to compare clauses to one another in the same way that whole sentences were with earlier versions of SARUMAN. This gives a value for each clause comparison on the same scale as a whole sentence.

Each clause is given a weight to signify its contribution to the meaning of the sentence based only upon its clause type. While there are no fixed values defined by Linguistics for these weights, the relative contribution of each clause is defined. Take the sentence:

"James killed Mike"

Logically, the verb clause is adding information to the understanding of the subject (James is a killer) and the object (Mike is dead).

Whereas the subject and object clauses only directly affects the verb clause:

"The car went to London".

Here, "went" is related to the action of the car and drive.

Finally, by definition other clauses are defined as subordinate (McArthur (ed.) 1991) and hence less significant.

Using the value of a noun clause to noun clause comparison as a base of unity, then a verb clause to verb clause comparison can be given a weight of 2 to reflect that it affects two clauses.

Because the value 2 is built from the multiplication of two weights, the weight of a verb clause is set to the square root of 2. A subordinate clause is taken as the reciprocal of the verb clause.

This gives a matrix of scores and weights identical in format to that which can be input to

the raw SARUMAN algorithm from chapter 6. Therefore, the final stage is simply to re-use the SARUMAN algorithm to produce the single number for the overall similarity.

8.6 Experiments

Because the cross-type comparison introduced in section 8.4 could potentially have affected the correlations from the versions of SARUMAN from chapter 7, the experiments from section 6.7 & 7.5 (without repeating the automatic disambiguation of meaning) were repeated using the automatic parser detailed in section 8.2 and cross-type comparison as described in section 8.4.

The core experiment is continued using the clause disambiguation as outlined earlier in this chapter in section 8.5.

8.7 Results

The increase in vocabulary combined with cross-type comparison means that the results for earlier versions of SARUMAN are slightly altered. Table 8.13 gives the summary information using the cross-type comparison for the models alongside the results for clause disambiguation.

	Pearson's	Spearman's
SARUMAN cross	0.319	0.334
Typed + cross	0.467	0.505
Human tagged + cross	0.583	0.468
Clause disambiguation	0.637	0.578

Table 8.13 Summary results for SARUMAN with cross-type comparison on the ten pairs dataset.

Both the basic model and the human tagged version show a small improvement in PCF over the correlations without cross-type comparison (table 7.3), although by definition in section 5.3.2, too small to be statistically significant. The relative improvement between the versions of the models remains and re-affirms the earlier findings.

Replacing the human tagging with the machine parsing for the part of speech gave similar results to those obtained without the inclusion of cross-type comparison. The difference in the parsed data was minimal between the human and the machine parsing, so the change in performance from the cross-type comparison can again be regarded as statistically insignificant.

The most important result is that the introduction of clauses has lead to a clear improvement over the previous version of SARUMAN, which only had type disambiguation, showing an improvement in Pearson's correlation of 0.17.

The fact that the introduction of clauses to SARUMAN gives correlation which is statistically significantly (over 0.05) greater than the use of human disambiguated meanings shows that clause disambiguation is essential to processing the ten pairs dataset accurately.

This is reflected in the graphical representation of the results for the automated parsing versions of SARUMAN shown in figures 8.2 - 8.4. The original free comparison is clearly improved by each step of the disambiguation. The value for the sentence pair differing only by punctuation shows how clause disambiguation can start to move the output closer to the human understanding of the meaning.

Figure 8.2 shows that cross-type comparison is very similar to SARUMAN (figure 7.5) and gives a too high a value for most sentences although with a weak resemblance to the shape of the human scores. Clearly, it is still a poorly performing model on the dataset.

Type disambiguation (figure 8.3) is still weakly performing and failing on pair 9 but does show a clear visible improvement over figure 8.2

The human selected definitions (from table 7.1) used with SARUMAN (figure 8.4)

showed very similar results with cross-type comparison as without in figure 7.7. It shows a closer correlation between the graphs than with type disambiguation alone.

SARUMAN with clause disambiguation (figure 8.5) is still showing significant differences between some of the values and the human rating but with a clear visual improvement from the disambiguation of type alone. It is the first purely automated version to be able to distinguish the meanings of the sentences in pair 9 as being non-identical.

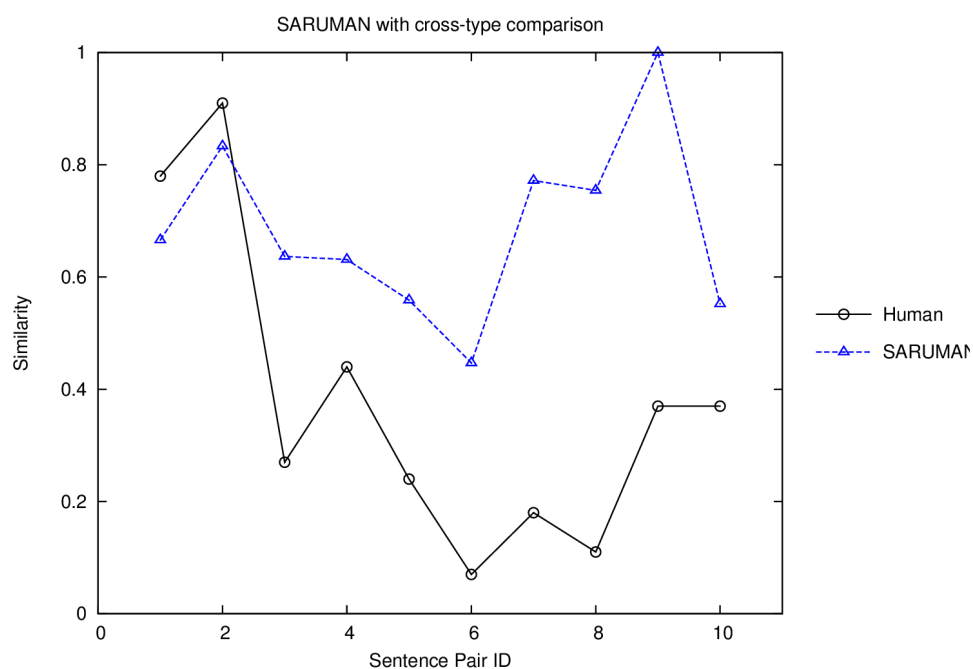


Figure 8.2: SARUMAN with cross-type comparison

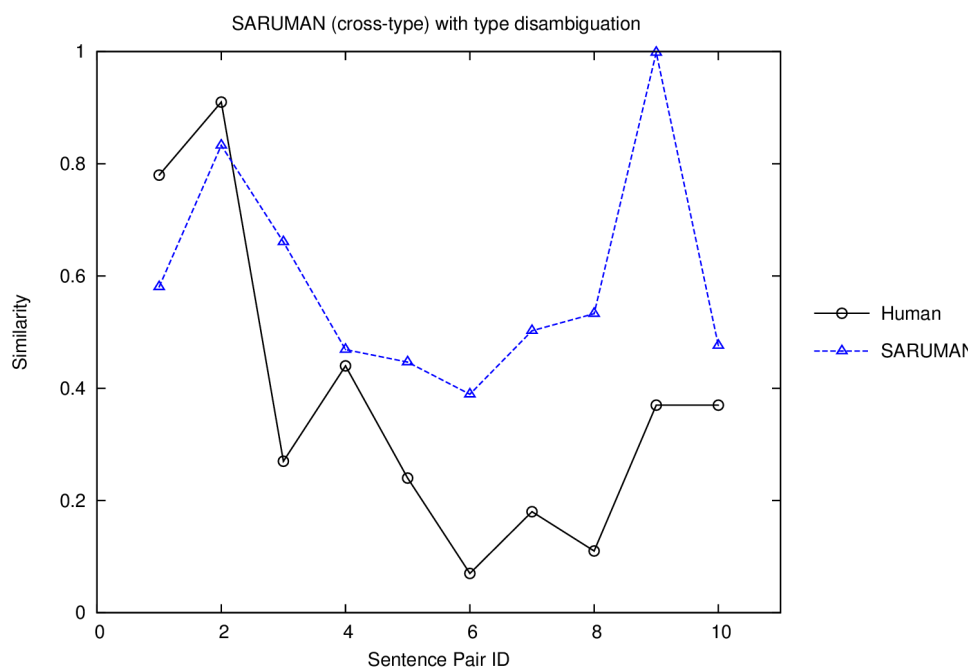


Figure 8.3: SARUMAN with type disambiguation and cross-type comparison

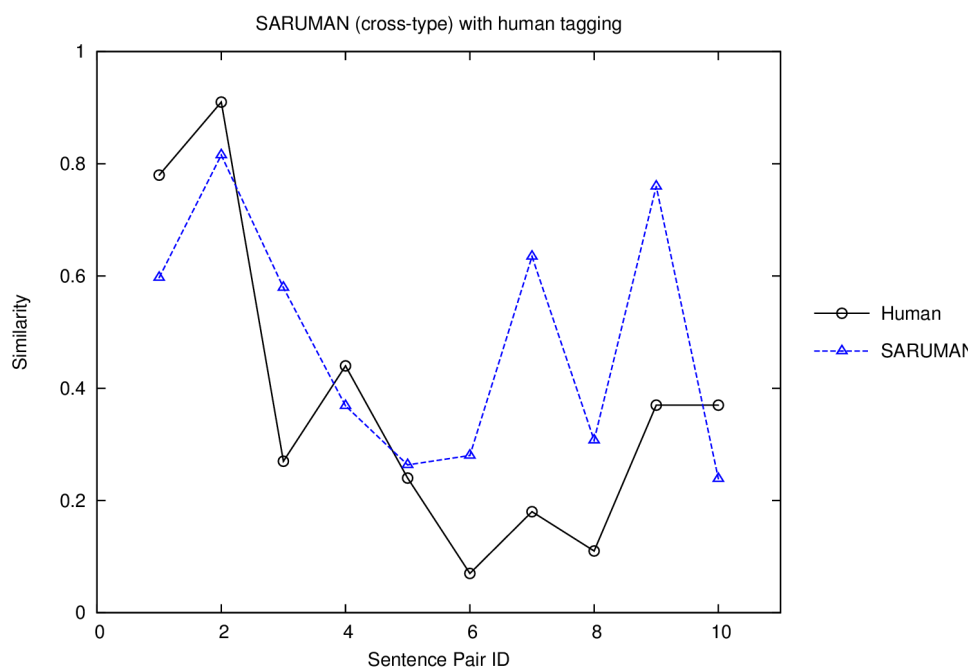


Figure 8.4: SARUMAN with human meanings and cross-type comparison -

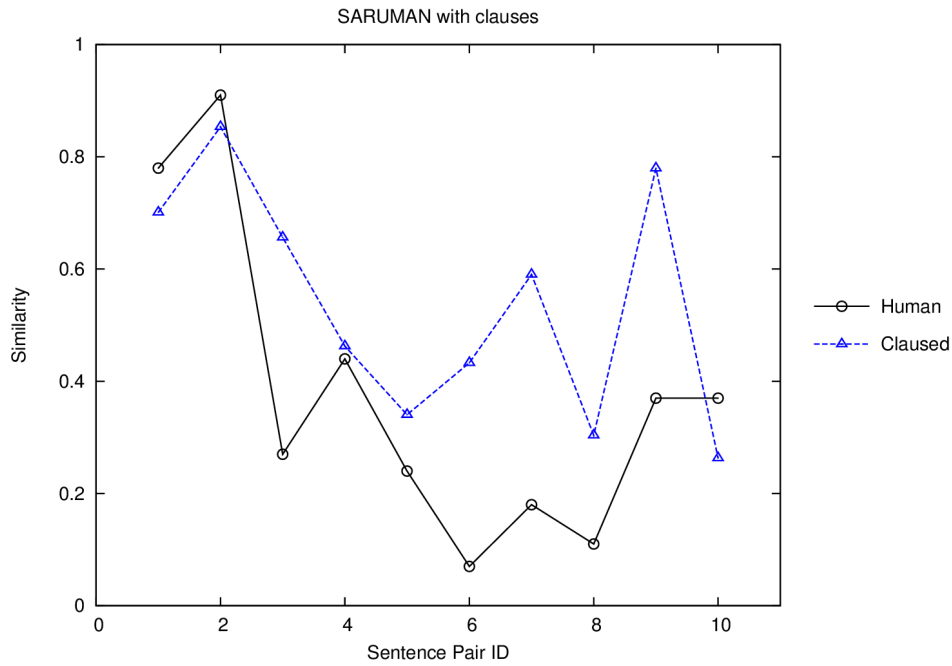


Figure 8.5: SARUMAN with clause disambiguation

8.8 Conclusions

This chapter again supported the premise being set out in the objective: (section 1.4) adding Linguistic components to sentence similarity could improve its performance. It was found that the inclusion of clauses gave a clear improvement over the mathematical version of SARUMAN with meaning disambiguation. The inclusion of clausal information even without the knowledge of the specific meanings was able to further refine the meaning comparison of sentences.

The novel sequential parser was able to perform strongly upon the ten pairs dataset and its ability to tag the sentence for both part of speech and clauses can be seen to greatly aid the performance of SARUMAN. While the parser with its specialised output has not been formally benchmarked, the approach of sequential parsing allowed for the efficient encoding of a large number of Linguistic patterns.

The automatic parsing to provide type disambiguation repeated the findings from using tagged data and a significant increase over the mathematical model was obtained, with a change in Pearson's correlation of almost 0.15.

Clause disambiguation enabled some of the more complex functional differences in the sentences to be accounted for when calculating the similarity and a strong improvement in correlation was again obtained at 0.17. Even without including meaning disambiguation itself, it was found that the model was able to exceed human meaning tagging by a statistically significant margin.

This shows that clause disambiguation can be an important feature to the similarity of a pair of sentences and that it was possible to include an implementation, which could be of benefit to a sentence similarity model.

The next stage of the development examines how the ideas can function together to form single concepts.

9.0 Combining Properties for Similarity

9.1 Introduction

The next stage to look at, for SARUMAN's development, is a more advanced form of word interaction that is interested in how meanings combine to form more complex meaning. For the basic clauses this involved merging the meanings of the individual words to be treated as a single meaning.

The main objective of this chapter is to make the preparatory changes to the Word Meaning Similarity module (WMS) in a manner that is conceptually consistent with the merging of meanings that relate to the advanced word interaction being added to SARUMAN, in order to produce SCAWIT in the next chapter.

The changes being examined in this chapter significantly increase the potential of SARUMAN but because the bulk of the knowledge source being used is unaltered from the cross-type comparison, it is not expected that the changes to the WMS would lead to a statistically significant improvement in the dataset.

The core conceptual change is to alter the WMS from using hypernym chains to representing the meanings as sets properties. The properties representation allow for the auxiliary verbs to be given a meaning structure that can be compared against other words. This is important for representing the meanings of the verb clauses with respect to the mood and tense. A set of properties based upon their logical relationships is given in section 9.2.

The handling of the cross-type comparisons was effectively already using a properties model but in these cases the common meaning was still being represented by a hypernym chain. While equation [6.1] was computationally the same as the original Li et al. (2003), it was only through its reconfiguration (to conform to the framework's requirements given by equation [4.2]) that it was possible to compare to structures with a different root node and come-up with a similarity other than zero. While technically conforming to [4.2] means

that the Li et al. (2003) algorithm could still be used for any structure where the common meaning could be found, there are other serious conceptual and functional issues that arise especially when looking at merging meanings, which are detailed in section 9.3.

Therefore, a word meaning similarity model that specifically designed to use properties of words (PoW) first presented in Pearce et al. (2011) was used instead, as the basis of the WMS. While most of the meanings remain with hypernym chains, it is not incompatible to use chains as sets of properties with the conceptual representation. This was the first time that properties representation has been used for a lexical approach. However, but because WordNet is sparse on properties and the knowledge source is still predominantly hypernym chains, the objective was to show that PoW could give comparable results to the Li et al. (2003) formula for SARUMAN for the versions with disambiguation by types and when divided into clauses.

Section 9.6 includes looking at the difference of merging two words in the Mitchell and Lapata (2010) dataset with the Li et al. (2003) versus the PoW model when using the current vocabulary.

The PoW formula would allow for the properties of the words in a single basic clause to be combined into a single set of properties equivalent to those of a single word (Pearce et al., 2011) were all the meanings stored as defining properties. To create an entire meaning representation for the properties of all the meanings currently handled by SARUMAN would be prohibitive and represent several thousand man hours. Using the hypernym chain dominate vocabulary therefore leaves much of the potential of the properties representation dormant.

This chapter also includes a provisional experiment where the properties have been created by-hand for just the selected meanings inside the ten pairs dataset. This has with the aim of showing that the deeper meanings from properties could outperform the current WordNet based hypernym structures. However, since this also includes meaning disambiguation and a very small sample of the vocabulary, this is only intended to show the motivation for wanting to change the representation of meaning, rather than being part of the core experiment.

9.2 Handling auxiliary and modal verbs

The auxiliary verbs can be subdivided into several groups (Strang, 1963) each one to be attributed properties based on five dimensions with an acronym of OPTIC and subdivided as follows:

- Outcome: finished - during - conditional
- Past: Past - present - future
- Task: finished - during - conditional
- Intent: Must - will - might
- Can: happened - can - could

Each dimension is currently being given three options: happened - ongoing - conditional. These dimensions and states are based upon the effect that the auxiliary and modal verbs can have on a transformational clause. The auxiliary verbs will affect the consequences of an action. Take an example of three forms:

The man killed the thief.

The man will kill the thief.

The man can kill the thief.

In the first example, the simple past tense means that the man is a killer and that the thief is already dead. The use of the future tense though that at the point of time of the utterance that the thief is alive. In the case of the “can kill”, the thief may never be killed by the man.

The premise being used is that when an event has already happened, then at some point in the past - it would have been happening and could have been described with the present tense, and before this in time could have been described with the future tense.

These states are given a weight between 0 - 3 and their values can be found in table 9.1. A value of 3 would indicate that all of the properties (past, present and conditions) apply (i.e. the event has happened), whereas 0 would mean that none of the attributes apply.

"Need to" and "have to" are treated the same as "must"; "ought to" as "should"; "may" as "can"; "might" as "could"; and "shall" as "will". Some of these parameters are obviously approximations of the meanings of the terms rather than an exact match to the terms. The trickiest is perhaps "used to" which due to a habitual nature refers to both the past but without the outcome having any effect.

	Outcome	Past	Task	Intent	Can
Present	2	2	2	2	2
Past	3	3	3	2	3
Used to	1	3	1	1	2
Should	0	0	0	3	1
Must	0	0	0	3	2
Can	0	0	0	1	2
Could	0	0	0	1	1
Will	1	1	1	2	2
Would	0	1	0	2	1

Table 9.1: O-P-T-I-C values for auxiliary verbs

The modal auxiliary verbs weights can be further altered when occurring in conjunction with the primary auxiliary verbs as shown in table 9.2. "Do" is regarded as not modifying the verb clause and any nuance to the meaning is currently not handled. Negatives do not alter these properties and are treated as a function for the whole clause and covered in more depth in chapter 14.

	Outcome	Past	Task	Intent	Can
Be + ing	0	*	1	*	*
Have been + ing	2	*	2	*	*
Be + ed	1	*	1	*	*
Have + ed / Have + been	2	*	2	*	*

Table 9.2: Effect of primary modal verbs and tense to O-P-T-I-C

The weights for the values are halved and rounded down before being used as an adjustment to *D1*, *C12* & *D2* for the main verb comparison. The information needed to identify the tense and modal verbs was already included within the parser.

9.3 Limitations of Li et al. Formula

It was shown that it was possible to use the properties representation with the Li et al. (2003) algorithm which is currently being used as the key part of the word meaning similarity module. However, there are still serious conceptual issues when considering using the formula for combining meanings.

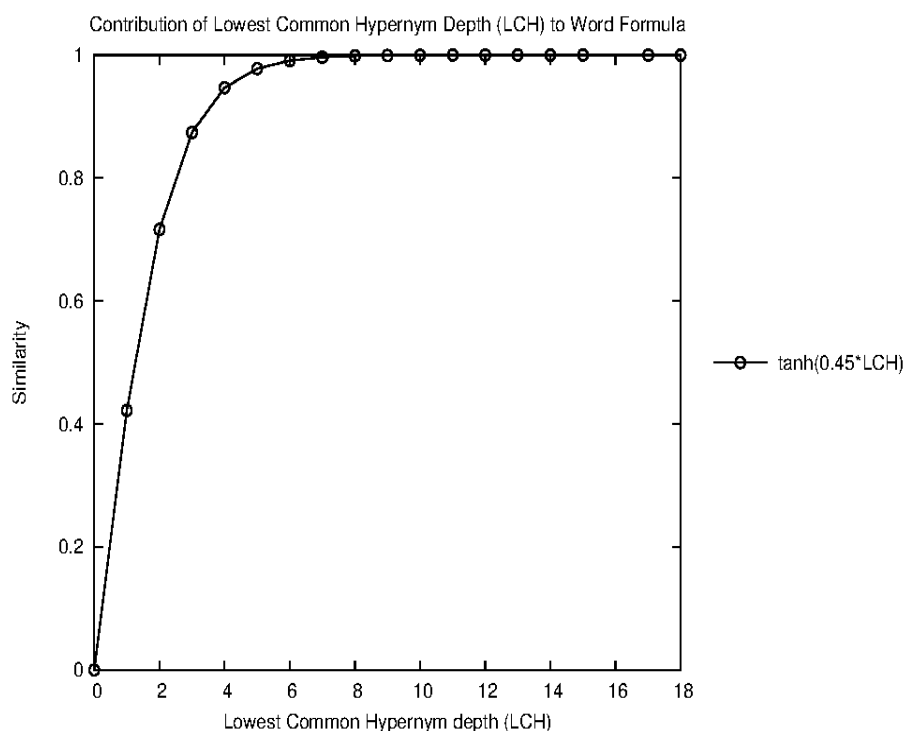


Figure 9.1: Contribution of common meaning to the Li et al. (2003) formula

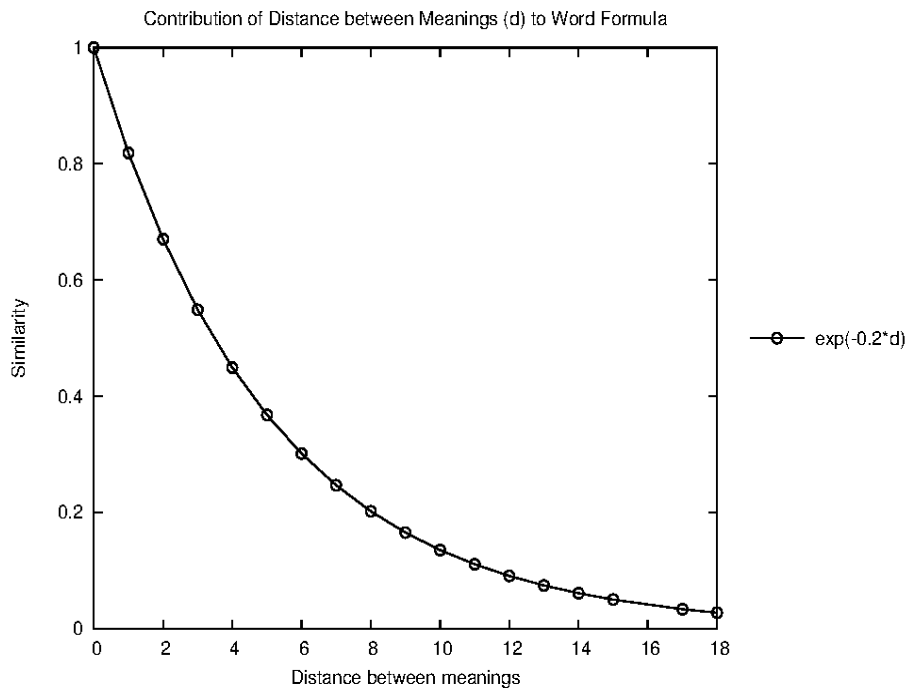


Figure 9.2: Contribution of Distance between meanings to Li et al. (2003) formula

Figures 9.1 and 9.2 isolate the effect of both the lowest common hypernym (LCH) and the distance between the pair of meanings for the Li et al. (2003) model and it can be seen that impact from the LCH quickly approaches its asymptote.

Consider two pairs of meanings which have the same distance but significantly different number of hypernyms between the meanings and the root node.

Cyclops - *crop* (a cultivated plant that is grown commercially on a large scale)

Bobcat - *tabby*

"Cyclops" and "crop" have a lowest common hypernym of "organism" whereas "tabby" and "bobcat" have a lowest common hypernym of "cat".

...organism - animal - giant - Cyclops

...organism - plant - crop

...cat - wildcat - lynx - bobcat

...cat - domestic cat - tabby

Now the meanings of "tabby" and "bobcat" are clearly more similar than "Cyclops" and "crop". This is reflected by WordNet because "cat" carries significantly more semantic information than "organism" as is represented by an extra 7 hypernym levels. However, the distance between the pairs of meanings is the same in both cases, which means if using the current word similarity module: "tabby" to "bobcat" score 0.3679 and "Cyclops" and "crop" score 0.3598 .

While there is a very small increase due to the extra 7 hypernyms, this is far less than the minimum difference from a single extra level of distinct meaning (adding 1 to the distance) that can happen within WordNet's current structure. Effectively, the Li et al. (2003) algorithm is heavily dominated by distance for all but the shallowest structure.

The Li et al. (2003) algorithm's minimal variance to the common structure, C12, makes it fundamentally conceptually incompatible with the idea of merging meanings.

9.4 Properties of Words Model (PoW)

It is technically still possible to mathematically use the Li et al. (2003) algorithm because of its compatibility with the parameters in the framework, as was shown from the cross-type comparisons which were effectively altering hypernym chain representation to a properties representation. However, in the strictest terms this means that it is no longer the same formula as its inputs are no longer the same format, despite being the same mathematical formula.

However, in order to allow for more conceptually consistent development of SARUMAN with further Linguistic concepts, an algorithm taken from Pearce et al. (2011), which was designed to use a properties representation structure, will be used for the Word Meaning Similarity module. The algorithm works on the basis that each of the meanings being compared can be represented by a set of definitive properties and that the overlap of these properties can be used to find the common meaning.

Where a definitive property is an attribute such as "grey" for an animal then this is

assumed that this implies that the animal also has a potential attribute of “colour” which has been given a value of “grey”. Hence, each property is viewed as having a weight of 2.

The definitive property is an attribute viewed as relevant to the meaning of the word, so the fact that an elephant has “molecules” is not regarded as definitive. People will normally learn and use the word “elephant” without having knowledge of molecules.

The algorithm to be used for the Word Meaning Similarity module will be referred to as PoW and given in [9.1] below. An alternative implementation for [4.2] is used where the common properties between the ideas is C12 and D1 the distinct properties for meaning 1 and D2 for meaning 2.

Where $D1 < D2$ ($D1$ and $D2$ are reversed)

$$\text{Similarity} = C12 / (C12 + 0.5 * (D1 + D2) * (1 + D1 / D2))$$

[9.1]

9.5 Example of a Clause Comparison

The following is an example for a noun clause - noun clause comparison:

"The big grey animal"

"An elephant"

A possible properties representation of each meaning could be expanded based upon the definitions as follows:

Elephant: big + grey + tusks + 4 Legs + tail + trunk + mammal + animal + alive

Animal: animal + alive

Big: big

Grey: grey

In their merged forms, the sets of properties for the two clauses become:

Clause 1: (alive - animal - big - grey)

Clause 2: (4 legs - alive - animal - big - grey - mammal - tail - trunk - tusks)

The properties being used to represent animal are very shallow and an alternative where these are given for a 4 properties version (double weight) will also be processed.

The individual comparisons become

Big to Elephant

$D1 = 0, D2 = 8, C12 = 1$

$PoW = 0.200 \parallel \text{Li et al. (2003)} = 0.085$

$D1 = 0, D2 = 10, C12 = 1$

$PoW = 0.167 \parallel \text{Li et al. (2003)} = 0.057$

Grey to Elephant

$D1 = 0, D2 = 8, C12 = 1$

$PoW = 0.200 \parallel \text{Li et al. (2003)} = 0.085$

$D1 = 0, D2 = 10, C12 = 1$

$PoW = 0.167 \parallel \text{Li et al. (2003)} = 0.057$

Animal to Elephant

$D1 = 0, D2 = 7, C12 = 2$

$PoW = 0.364 \parallel \text{Li} = 0.177$

$D1 = 0, D2 = 7, C12 = 4$

$PoW = 0.533 \parallel \text{Li et al. (2003)} = 0.233$

This would give values from SARUMAN for the 2 property animal:

SARUMAN - $PoW = 0.426 \parallel \text{SARUMAN - Li et al. (2003)} = 0.244$

and for the double weight:

SARUMAN - $PoW = 0.477 \parallel \text{SARUMAN - Li et al. (2003)} = 0.284$

The merged versions become:

"The big grey animal" to "An elephant"

$D1 = 0, D2 = 5, C12 = 4$

PoW = 0.615 || Li et al. (2003) = 0.348

$D1 = 0, D2 = 5, C12 = 6$

PoW = 0.706 || Li et al. (2003) = 0.365

This would give values from SARUMAN for the 2 property animal:

SARUMAN - PoW = 0.615 || SARUMAN - Li et al. (2003) = 0.343

and for the double weight:

SARUMAN - PoW = 0.706 || SARUMAN - Li et al. (2003) = 0.386

While it can be seen that treating the words in the clause as a single entity improves the performance, regardless of whether PoW or the Li et al. (2003) algorithm is used, the output of the Li et al. (2003) algorithm still scores far lower than the actual overlap in meaning would suggest. So it would suggest that the PoW model is performing better than the Li et al. (2003) algorithm for properties, as was expected from the discussion in section 9.2.

9.6 Experiments

The new version of the word similarity module using the properties of words (PoW) formula given in [9.1] potentially altered the findings from the key experiments with SARUMAN from chapters 6-8.

The aim is to reproduce the 3 main benchmarking experiments of: SARUMAN, SARUMAN with type disambiguation and SARUMAN with clauses but using the PoW formula in place of the Li et al. (2003) formula. The next stage of development of SARUMAN wants to expand upon the conceptual ideas using properties and requires that the WMS is conceptually compatible with its development.

The aim of these experiments was to reaffirm that each version of SARUMAN is still giving statistically significant improvement over the last. The experiment with SARUMAN and clause disambiguation can also include the auxiliary and modal verbs, as outlined in section 9.2. It is not expected that these changes will significantly alter the performance as it is a sparse feature in the ten pairs dataset. The results of these experiments can be found in section 9.8.

Before these experiments are given, though there is another issue in that essentially that the meaning structures being used are still based upon the hypernym chains from within WordNet (Feldbaum (ed.), 1998). The Li et al. (2003) model was specifically tuned for these structures whereas PoW was based upon sets definitive properties for each meaning (Pearce et al., 2003). It was shown in section 9.2 that the Li et al. (2003) formula was conceptually flawed for merging ideas but it has not been shown that PoW could make a practical replacement using the existing knowledge base.

Therefore, an experiment to compare the to algorithms performance as part of SARUMAN on very simple units of language where pairs of words are combining together, were carried out using the Mitchell and Lapata (2010) dataset (described in section 3.2), which uses pairs of words with known parts of speech. This means that most of the elements of SARUMAN other than the WMS and merging of words are isolated. The objective was to show that similar results between Li et al. (2003) and PoW (treating each node in the hypernym chain as a property) could be achieved. Therefore, indicating that even with a suboptimal knowledge base that a properties word formula could be used. These experiments are given in more detail in section 9.7.

One final experiment is included in this chapter, wherein a set of properties was constructed for each word in the ten pairs dataset based upon their current context. This was then used as a provisional experiment to indicate the possible potential of SARUMAN when coupled with a full properties representation. Since only a single meaning for each form of the words in the ten pairs dataset was given issues of disambiguation, they are removed. Therefore, a more accurate performance would be expected. However, it is also

the case that the issues with automatic disambiguation and human tagged meanings from chapter 7 would likely be reduced with a full properties knowledge base. The keywords used by automatic disambiguation would already be present with less noise and more significantly, the comparison between a meaning close to the intended meaning but wrong, would give a closer match in similarity as the properties representation is deeper. However, it would be prohibitive to replace the entire knowledge base when WordNet was a collaborative approach taking many 1000s of man hours .

9.7 Comparing Word Models with Merging

It can be seen that with a properties representation and the associated merging that the PoW (properties of words) algorithm (Pearce et al., 2011) has the potential to be more representative than the Li et al. (2003) model. However, the knowledge base is still remaining the same and so a test was made to determine whether to replace the Li et al. (2004) formula with PoW in the Word Meaning Similarities module (WMS) and obtain similar results, even using the hypernym based meanings.

The Mitchell and Lapata (2010) dataset (section 3.2) comprises pairs of words, first classified by type: noun-noun; adjective-noun or verb-noun, and then as low, medium or high similarity. Both PoW and the Li et al. (2003) method were tested using the SARUMAN algorithm and the combination of clauses as outlined in the previous section. Since the parts of speech were already predefined, there was no parser involvement needed to rate the sentences.

Since the participants were asked to classify the words into equally sized groups from sets of 108 pairs to yield high, medium and low, it is not possible to assign approximate values to each of the levels. This is because, the decision as to which set a word was placed was not dependent upon an intrinsic level of similarity, only a relative placement. Therefore, the only option is to optimise thresholds on the output of the data.

WMS Method	Adj. - Noun	Noun - Noun	Verb - Noun
PoW	63 / 108	57 / 108	59 / 108
Li et al. (2003)	63 / 108	55 / 108	61 / 108
Li et al. (2003) Cross-type	64 / 108	55 / 108	61 / 108

Table 9.3: Classification accuracy for SARUMAN for different word similarity modules on Mitchell and Lapata (2010) dataset

The two word models yielded very similar results for the accuracy of classification as shown in table 9.3. Both are doing significantly better than the average from random chance (which would have been 36 / 108) but slightly lower than possible reasonable expectation. Alongside earlier issues discussed on the limits of the vocabulary, there are several pairs where the human raters have confused association with similarity. For example: "Elderly woman - Black hair" was scored as medium when it should be low.

The closeness in performance was sufficient to judge that continuing with PoW instead of Li et al. (2003) is possible without significant deterioration in performance, even with the current meaning structure rather than a properties model. The PoW model would still allow for the inclusion of properties alongside the hypernym representation.

9.8 Results

SARUMAN was re-run both with and without type disambiguation to confirm that the previous observed improvement was still resultant. Then the clause disambiguated version of SARUMAN was re-run including the modal and auxiliary verb properties, via OPTIC. Despite a slight fall in the correlation using the basic version of SARUMAN and the disambiguation by type, there was an improvement from the clause disambiguated version.

The change at 0.02 was too small to be deemed statistically significant. However, the inclusion of properties for the modal and auxiliary verbs could well be causing a small improvement on the dataset.

Model	Pearson's	Spearman's	R.M.S.
SARUMAN - raw	0.207	0.213	0.473
SARUMAN - type disambiguation	0.419	0.395	0.385
SARUMAN - claused + OPTIC	0.659	0.596	0.280
Properties by hand	0.810	0.809	0.207

Table 9.4: Summary information with PoW the ten pairs data

Also a by-hand experiment was done by setting groups of properties for each of the meanings encountered in the ten pairs dataset, in the same manner as was shown for "elephant" in the example of merging. This was done to give a possible indication of the properties approach with SARUMAN, since it is architecturally more consistent with the combinations being done by SARUMAN with its cross-type comparison and PoW.

The comparisons were still restricted to the clauses as was the case for the latest version of SARUMAN with clauses, but the properties were used with PoW for comparison. As can be seen in the summary metrics in table 9.4, a very strong correlation is achieved for the properties by hand model but effectively this is include much of the benefit obtained from using human meaning selection as seen in the experiments in chapter 7, rather than simply the changes in conceptual representation alone.

The values for the properties by hand method show in table 9.5 are mainly fairly similar to the scores for SARUMAN using the WordNet meanings except on pairs 5, 6 and 8. Disambiguation of meanings has allowed many of the unwanted weaker connections being found by SARUMAN, to be excluded. Additionally, the combination of clauses when there was an adjective and noun were being combined showing a level of overlap for the adjective due to the shallow structure in WordNet, whereas with the properties representation, this is currently adding zero similarity.

ID	Human	SARUMAN	SARUMAN typed	SARUMAN claused OPTIC	SARUMAN Properties by hand
1	0.78	0.695	0.631	0.705	0.91
2	0.91	0.873	0.885	0.904	0.939
3	0.27	0.746	0.580	0.744	0.607
4	0.44	0.544	0.759	0.552	0.480
5	0.24	0.729	0.544	0.529	0.172
6	0.07	0.577	0.606	0.271	0.117
7	0.18	0.845	0.507	0.459	0.084
8	0.11	0.818	0.574	0.551	0.100
9	0.37	1.000	1.000	0.739	0.700
10	0.37	0.659	0.598	0.268	0.181

Table 9.5: Numerical results for SARUMAN with PoW on the ten pairs dataset

This was of course a crude method and the properties selected included human meaning disambiguation. The properties chosen might be different when considering the meanings of every possible word as would be the case with a complete knowledge base using properties. Nonetheless, it does suggest that the properties representation has potential to outperform the hypernym model. The graphical representations in figures 9.3-9.6 clearly show the results are improving in a visible manner.

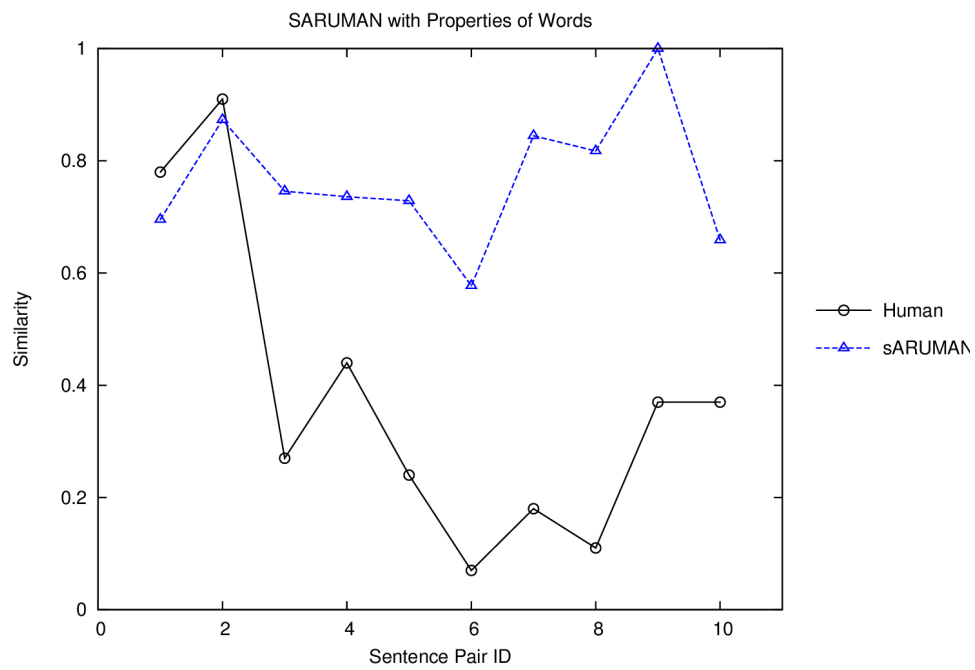


Figure 9.3: SARUMAN with PoW

The properties model makes little difference to the performance of SARUMAN without disambiguation. From figure 9.3, it can be seen that the values are significantly out, although there is some resemblance in shape of the curves formed.

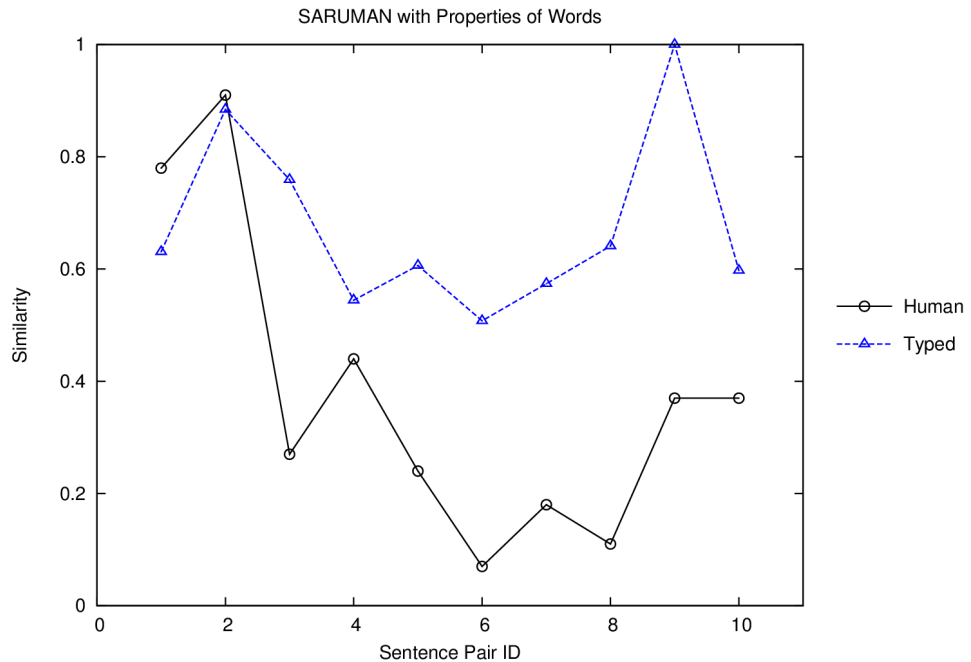


Figure 9.4: SARUMAN with PoW and type disambiguation

As before, with PoW the disambiguation of type (figure 9.4) moves the values significantly closer to the human scores, although the absolute values are still some distance from the human values.

The inclusion of clausal data shown in figure 9.5 as in the previous chapter, gives a marked improvement over the performance with just type disambiguation and gives a visibly stronger performance particularly with pairs 6 & 9, than seen in figure 9.3.

The Disambiguation of clauses combined with the human constructed properties in figure 9.6 shows a much closer shape and level to the human values than when using the WordNet based meanings. The combination of direct human knowledge with clausal information shows a strong ability to correspond to the human data, with 3 pairs being significantly higher than the human rating but many remain close to the human values.

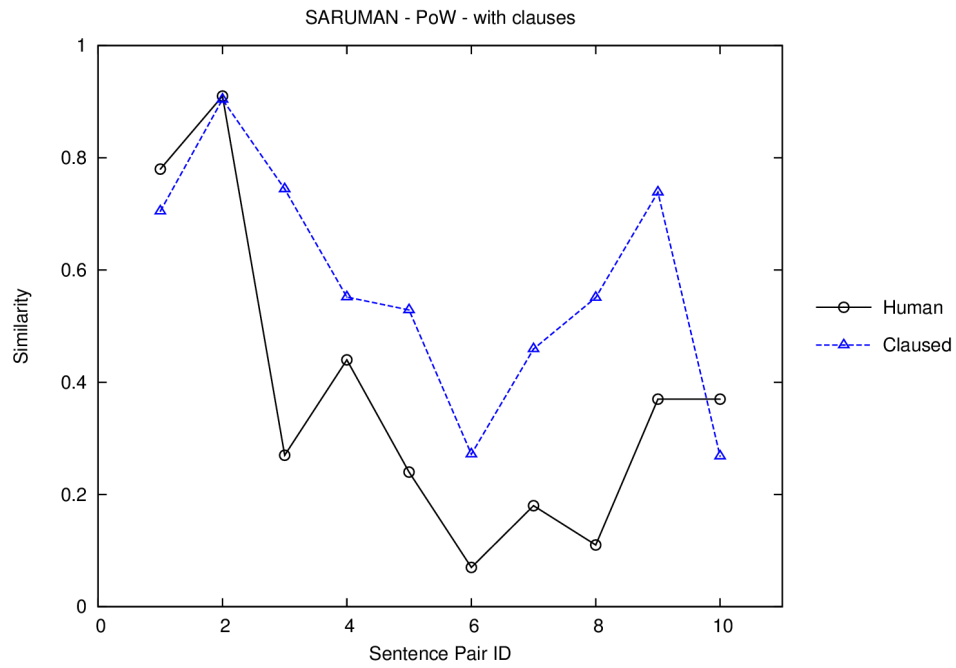


Figure 9.5: SARUMAN with PoW, OPTIC and clause disambiguation

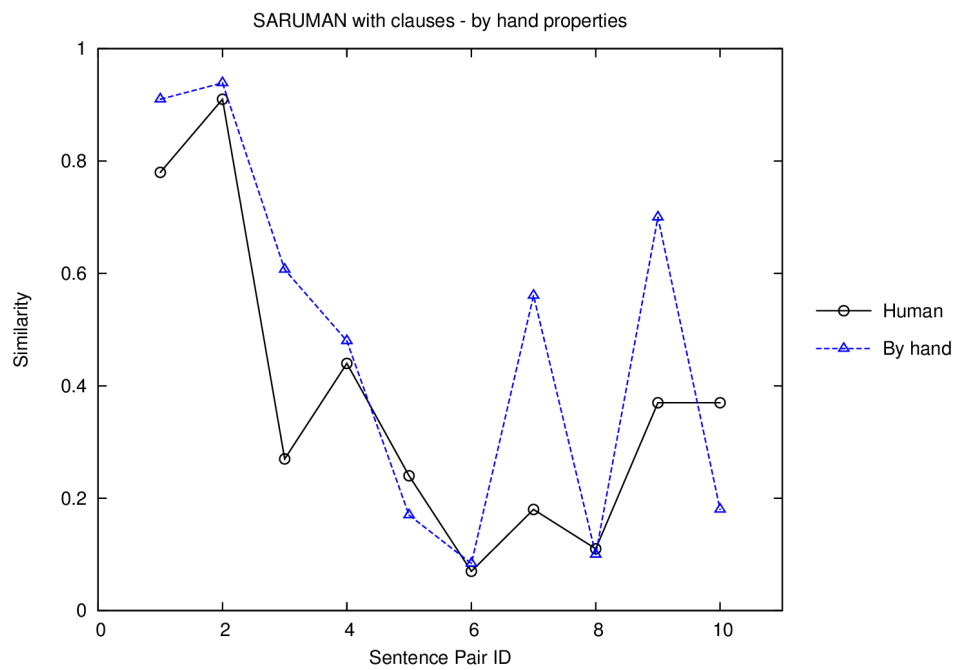


Figure 9.6: SARUMAN with clause disambiguation and human constructed properties.

9.9 Conclusions

This chapter changed the word similarity algorithm to one which is better suited for word interaction, when ideas are being combined together. The algorithm showed comparable results to the previous hypernym based word algorithm. Repeating the early experiments with the new word algorithm replicated the relative improvement in performance that had already been established. Although there was a small improvement in correlation from the combination of the new word similarity model and increasing the vocabulary to handle the modal and auxiliary verbs as part of the verb clause, it was not large enough to be regarded as clearly statistically significant.

The improvement of 0.02 in Pearson's correlation is likely in part the inclusion of the OPTIC data, but the influence on the dataset is too small to state that this is a definitively superior model. The by-hand model however, does show clear improvement over the previous version. Even considering the 0.10 change in correlation that human disambiguation had contributed over type tagging alone in chapter 6, the by-hand data still gives significant improvement on its own.

While from a computational perspective the difference between the version of SARUMAN in this chapter compared to the last is not significant, from a Linguistic perspective, it is highly significant. The reason is that the conceptual inconsistency arises between the steps that are being implemented and the future extension to the model, using the framework that needs to be done in order to allow for the merging of clauses which follows in the next chapter. So in conjunction with the by-hand results the version of SARUMAN using the PoW algorithm can be considered superior even when using the current ontological model which was used with the Li et al. (2003) algorithm.

10.0 Advanced Word Interaction

10.1 Introduction

This chapter introduces advanced word interaction to SARUMAN, to create a new version called SCAWIT. Whereas so far the structural information of the sentence has been used to ensure like for like comparisons, there are occasions in English when highly similar meaning can be expressed with very different grammatical structure.

The last chapter included a change to SARUMAN in the Word meaning similarity module (WMS) and conceptual representation of the meaning structure from the knowledge base. The changes to use the Properties of Words formula (PoW) is what allows the advanced word interaction between merging meanings to be included.

The results for SARUMAN in the last chapter will be compared against the new model, SCAWIT, concluding the core experimentation. Again, it is the objective of this chapter to find if a statistically significant improvement in correlation can be obtained from allowing meanings to be merged.

This conceptual change potentially allows for the meanings of words within a basic clause to be merged and combined as a single meaning. The basic clauses can come together to form more complex clauses. While the clauses define the functional structure of the sentence and how the meaning combines, comparison using clauses does not necessarily lead to the closest comparison between a pair of sentences. This is because English is capable of describing highly similar meanings using different structural forms.

The comparison between basic clauses has an analogy to the simpler units within the knowledge base (noun clauses to nouns) that could be used. Complex clauses formed of more than one basic clause have no such convenient relationship. If a situation arises where, due to the similarity, that the most appropriate comparison is between a basic clause and a complex clause then in order to enable the comparison, it could be that only part of the complex clause should be used for the comparison.

The part of the complex clause needed would be its basic clause that relates closest to the function of the basic clause with which the comparison is being made. combined with the relevant meaning from the remainder of the complex clause.

The remainder of this meaning is essentially part of its other basic clauses that can be combined with its matched basic clause to still form an equivalent meaning structure. This is effectively achieved because of the conceptual change to a properties model in the WMS in the last chapter. The situation arises where some of the properties of the object clause are combined with its verb clause, to enhance the meaning.

A distribution of the object clause is made so that part of it is merged with its verb clause and the remainder compared as its basic noun clause. This merge is accomplished by the concept of triangulation described in section 10.2.

Before triangulation can happen, the sentence needs to be categorised into its basic clauses and identify which basic clauses can be combined together to form more complex clauses. This then is followed by an alignment stage so that the clauses are coupled with the structure in the other sentence that is performing the same function. The alignment is made around the verb clause and detailed in section 10.3.

The alignment stage basically allows the weights matrix to be converted to its two vectors: one for each sentence with a value and weight for each of its clauses. The vectors are modified based upon the possible complex clauses and triangulations. This introduces both an order of calculation and a change of the weights to SARUMAN that are given in sections 10.4 & 10.5.

Although, the concept of merging was given based upon the properties model enabled by the changes in chapter 9, the knowledge base is still essentially coming from hypernym chains. This means that an implementation of triangulation is needed based purely upon the comparisons between the basic clauses. This is used to estimate how the distribution and the merging should be combined when triangulation is judged as advantageous. Section 10.6 details the algorithm and includes some basic examples.

These changes have made some substantial changes to the conceptual manner in which SARUMAN handles the similarities in its algorithm module. These were considered distinct enough to warrant this version of SARUMAN be given its own acronym of SCAWIT (SARUMAN with Clauses, Advanced Word Interaction and Topic). Essentially in many respects it remains the same algorithm and in many cases no triangulation may happen for a sentence pair.

The experiments include the continuation of the core experiment and SCAWIT is tested on the ten pairs dataset and compared to the previous versions of SARUMAN with PoW shown in chapter 9. In addition to the approximate method of triangulation that was used, as with the last chapter, the same properties by hand representation of the sentences is used. This time with the SCAWIT algorithm as an indication of its potential and to show whether the SCAWIT algorithm was showing an improvement when using properties compared to the clausal version from the last chapter.

10.2 Triangulation

The basic clauses combine together to form more complex clauses such as, with the predicate or main clause. The clauses describe the functional relationship of how the meaning combines together to form more complex meanings. Because the meanings function in overlapping units, it means that the similarity comparisons using direct clause comparisons, does not always directly describe the manner in which a person would interpret the meaning.

It is possible for highly similar meaning to be expressed whilst having different grammatical structure.

For example "hammered" and "used the hammer" have similar meaning but the first is a single verb clause, whereas the latter is a predicate with a verb clause and a direct object clause.

Basic clauses can directly correspond to the meaning related by a single word such as a noun clause and a noun. This meant that there was already a conceptual mechanism in place to compare the two units.

Using a description of the clauses as properties, has the potential to combine the two basic clauses into a single clause which is also simply a set of properties. This can result in the loss of detail from the meaning of the clause. If all the clauses were indiscriminately combined together as their grouped properties, then this would result in the situation where the word interaction was lost from the meaning and only a bag of words remain.

However, the idea of properties still enables the idea of merging the clauses in the same manner (as was accomplished with comparing noun clauses to nouns). In contrast to the previous properties model, often only a partial merging is wanted.

From the initial example, it would be possible to consider the verb clause - verb clause comparison "hammered" - "used" and then merge part (or in this case all) of the meaning of "hammer in with "used".

This merging involves a pair of clauses merging in order to improve the comparison to a single clause. This forms a triangle of clauses to be compared.

10.3 Alignment

The first stage is an alignment of the two sentences being compared. An alignment centred upon the verb clauses is made between the sentences. Considered in blocks containing a single verb clause, the longest pattern that arises in a single main clause comparison is:

Prep. - Subj. - Subj Prep. - Verb Clause - Obj. - Ditrans. - Prep.

The clauses are aligned so as to be orthogonally coupled with a like-for-like clause in the other sentence. As often there will not be a directly corresponding clause in the other sentence, a null clause is used whenever there is no directly corresponding meaning in the

other sentence.

A null clause can be processed in the same way as a regular clause only it is given a weight of 0 and a similarity of 0 between itself and any other clause.

10.4 Order of Calculation

Only certain clauses can function together as a pair in order to form a more complex clause. It is possible that more than two clauses can combine to form a more complex clause. The order in which merges are performed can alter the effect upon the similarity of triangulation.

When multiple clauses can interact with the same clause, it is necessary to introduce an order of calculation that reflects their Linguistic function. This is mainly because of the Linguistic relationship between the clauses when they combine to make a more complex meaning.

The following order of combination are adopted:

Verb clauses + Object clauses

Subject clauses + subject prepositional clauses

Subject clauses + Predicate

Verb clauses + prepositional clauses

Object clause + prepositional clause.

Main clause + subordinate clauses

The order of the combination is dictated by the units that are combining together. The predicate is composed of the verb clause + object clause. So this entity needs to be constructed prior to combining the Subject + predicate. Likewise, the subject clause and its conditional prepositional clauses need to be found prior to this combination. The main prepositional clauses can be thought of as functioning on the sentence as a whole and combine with the verb clause. So the order of the calculation is determined from which

clauses can be needed to construct the more complicated ones in increasing order of complexity.

Consider the following comparison of two fragments:

“the middle of the target”

“Bull's eye”

The first example can be divided into a noun clause + a prepositional clause, whereas the second example is just a noun clause for linear alignment: a null clause needs to be included.

The first step is to merge the basic clauses to form the larger noun clauses so for example:

"the middle" + "of the target"

can be compared against a single noun clause as if it had the same structure with a null clause in the place of a prepositional clause.

"Bull's eye" + NULL clause

Then the predicate has to be found before considering the other comparisons, as this effects the meaning of the verb which is the dominant contributor to the transformational effect of the sentence. Once the triangulation has occurred, cross clause comparisons can still be considered as before, but with the new weights and similarity scores for the like-for-like comparisons.

10.5 Weights

The clause weights remain in the same ratio as before as if there were a PSVO (prepositional clause, subject clause, verb clause and object clause).

Figure 10.1 illustrates the weights for a PSVO to PSVO comparison and shows how the individual weight for each clause was multiplied to give the overall relative impact of a particular comparison. The main diagonal gives the like-for-like comparison weights.

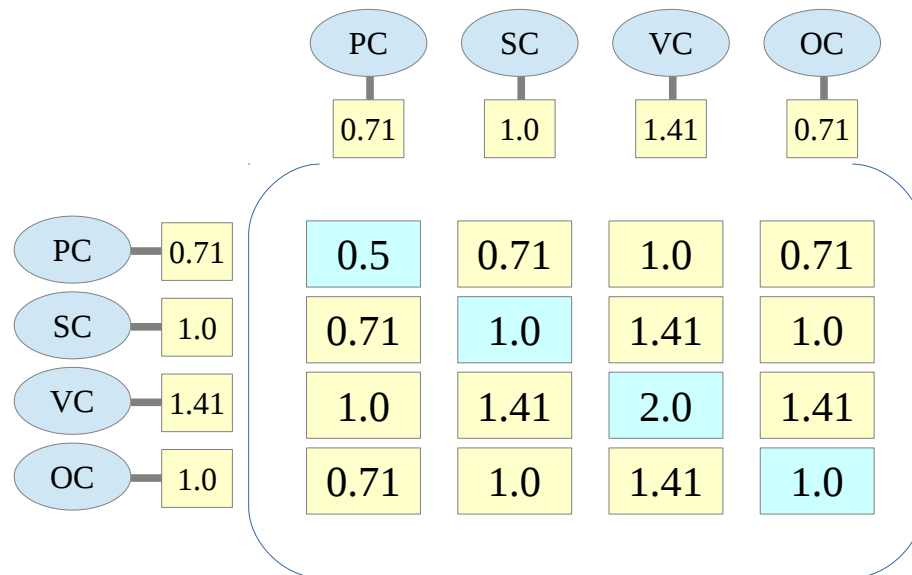


Figure 10.1: The importance weights for two PSVO clauses using the weights for SARUMAN (section 8.4).

The alignment of clauses is possible for a pair of main clauses with a PSVO, through the inclusion of null clauses. These clauses would have a similarity score of 0 and so would add nothing to the overall similarity. Figure 10.2 shows how the used weights for an aligned pair of clauses could be represented.

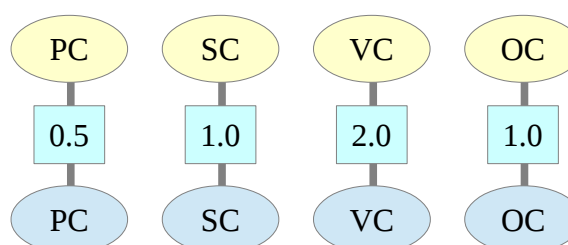


Figure 10.2: The importance weights for two PSVO clauses aligned

The weights for the aligned situation were originally formed from a multiplication as seen

in figure 10.1. This signified the importance of the comparison to the overall similarity. For triangulation though, the contribution of the individual clause (as opposed to as part of a pair of clauses) to the overall meaning of the sentence needs to be used. This is done by assuming that each of the aligned comparisons contributes equally in a linear fashion leading to the situation shown in figure 10.3 below.

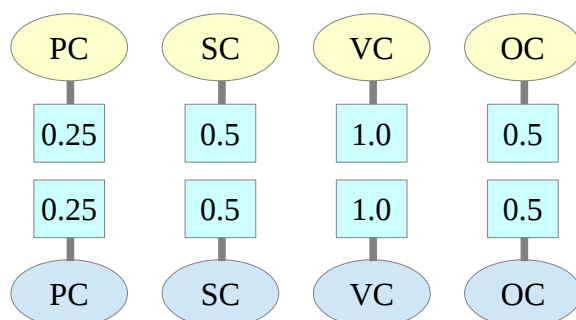


Figure 10.3: The importance weights distributed to individual clauses for two PSVO clauses aligned

The actual absolute value of the weights are not important to the algorithm, only the ratio giving the relative weight matters. It was necessary to make the change to distribute the weights, as the combination requires a linear distribution of weights. Subordinate PSVO clauses are still adjusted as before though with a multiplier of root 2.

10.6 SCAWIT Algorithm Module

The way in which the information is now being combined alters the algorithm of SARUMAN, as it has to adjust the similarity scores and weights to account for triangulation. This adjustment is significant but essentially still maps onto the same algorithm as currently being used by SARUMAN.

Two pairs of corresponding aligned clauses are affected from the action of the triangulation. Firstly, the pair into which the basic clause is merged has its contribution to the overall similarity increased, through blending the cross-comparison with the unmatched part of the similarity of the target pair. Second, the pair where clause being merged was originally aligned has its similarity reduced.

Consider two pairs of clauses (i.e. two predicates) that are being considered for triangulation A-C and B-D. (A is the verb clause and B the Object clause in sentence 1)

A		B
	X	
C		D

There are four possible triangles which need to be considered for triangulation: A-B-C, A-B-D, C-D-A and C-D-B. Assume that it has been decided that A-B-C is going to undergo a triangulation merging, so that part of the meaning of B, B' , will be combined with the clause A and the remainder, B^* , will still pair with D.

A - B'	B^*
/	
C	D

Now the affect of the substituting $A + B'$ for A gives a new pairing A' -C with an increased similarity and weight.

With a properties model this can be performed as a direct merging as was seen with the example for merging words within clauses in the last chapter.

So if comparing:

"used + the whip"

"stabbed + his arm"

With property sets as follows:

$use = (use + function + tool)$

$whip = (weapon + whip)$

$stabbed = (use + function + tool + knife + weapon + damage)$

B' becomes simply "weapon" and B* is just "whip".

Then A' is $(use + function + tool + whip)$

Using PoW, A-C was 0.667 and A'-C becomes 0.800 and the weight of the pair needs to increase to reflect the increased semantic contribution to the sentence now that part of the word clause is included. In this case, half of the original weight for the noun clause "the whip" is transferred from B-D to A'-C.

However, since there is not currently a complete properties model, SARUMAN needs to make an approximation which is achieved through considering the similarity B-C ("the whip" compared to "stabbed").

Essentially, to replicate the merged properties model would require being able to make a tertiary comparison. To represent the comparison would require the 7 possible overlapping and distinct areas such as would be obtained from a Venn diagram. Even were the similarity between A-B to also be used then there would be an unknown parameter. This means that an estimate of how to merge the similarities has to be made.

Matters are further complicated since the cross-comparison (B-C) might be using a different meaning for C than was the case for (A-C).

An approach that views the similarity score as the ratio of common meaning to total meaning will be presented (unmatched difference, U, + similarity, S).

If similarity between A and C is ' f ' and for the cross comparison between B and C is ' g ' (so $A-C = f$ and $B-C = g$) then the ratio of the meaning that is being contributed from B to get B' is simply ' g ' as the remainder of the meaning remains in B*. Although this seems intuitively a high contribution, because ' g ' is never going to be unity due to the commutativity and C will have many distinct properties, the adjustment is still smaller than

could be possible with the result from a merged value with the properties model.

U is $(1 - f)$ and is reduced by a ratio of g which adds to S.

The new similarity A'-C is expressed as:

$$g * (1 - f) + f$$

Where f = similarity (A-C) and g = similarity (B-C) .

[10.1]

The B*-D also has to be reduced. If B-D had a similarity of k then the remaining unmatched and similarity become reduced by the ratio of the part of the meaning removed (assuming independence between B-C and B-D).

$$U = (1 - k) * (1 - g) + k * g$$

$$S = k * (1 - g)$$

$$S + U = 1 - g + g * k$$

So B*-D becomes:

$$(k - k * g) / (1 - g + g * k)$$

[10.2]

This is consistent with the required increase to A-C becoming A'-C and B-D becoming B*-D.

The final stage of the algorithm is accounting for the weight change. Weight of A' becomes:

$$(Original\ weight\ of\ A) + g * (original\ weight\ of\ B)$$

[10.3]

And B^* becomes

$$(1 - g) * (Original\ weight\ of\ B)$$

[10.4]

The total weight of the sentence is left unaltered. The initial matrix of clause comparisons is updated for the new values of A-C with A'-C but with B-C changed to B^* -D.

The one critical stage not yet described is determining when the merge should take place and which of the four combinations should be used. This is decided through a simple set of greater than comparisons.

Firstly, the cross comparison needs to be larger than either of the orthogonal like-for-like comparisons and larger than the threshold used for the position similarity (> 0.2).

If that condition is met then the larger of the diagonals is used and the merging happens from the weaker matching clause to the stronger.

So if $A-C = 0.8$, $B-D = 0.4$ and the diagonals $A-D = 0.5$, $B-C = 0.6$ then because $B-C > A-D$ and $B-C > 0.2$, the merge happens from A-B-C or B-C-D. Because $B-D < A-C$ then the merge happens into A-C so the triangulation A-B-C will be used.

After processing all of the merges, a single value is obtained as before from the remaining matrix.

When there are multiple verb alignments this then produces a weighted matrix of values identical to that with the clause disambiguated version of SARUMAN and again the algorithm is simply used to produce the overall similarity. The weights for subordinate verb clauses are again using the subordinate weight of half root two (~ 0.707).

10.6.1 Limitation

Triangulation is an approximate method and it is possible in some circumstances that triangulation could lead to an overestimate of similarity. Such as where the different structure allows for additional clauses to affect the meaning of the combined clause.

So while “used the whip” can mean the same as “whipped”, were it to appear in a sentence such as “He used the whip to hold open the door”, the added clause “to hold open the door” has altered the meaning of “whip”. Its usual function is ignored and instead is simply being used as a generic object.

For an overestimate of the above type to exist there must have been an intrinsic difference in the structure of the other clauses to change the functional interpretation of the sentence. This means that there must still remain a difference in the meaning which would be still be detected so the magnitude of the error is less than from the ignoring the contribution of merging.

10.7 Experiments

The implementation of SCAWIT, as described in sections 10.2-10.6, was tested on the ten pairs dataset and compared to the version of SARUMAN with PoW given in the last chapter. This is a continuation of the core experiment to examine whether the inclusion merging clauses would lead to an improved performance.

Although, as outlined in chapter 9 prior to the introduction of PoW into the Word Meaning Similarity module for SARUMAN, the Li et. al (2003) formula has conceptual issues with merging, a version with SCAWIT using Li et al. (2003) is included for completeness. As the majority of the knowledge base is still using structures equivalent to hypernym chains, it is still expected for both versions of SCAWIT to show improvement.

The final experiment being presented in this chapter again uses the same sets of properties as had been used in chapter 9 for the by-hand version of SARUMAN including clause disambiguation. In section 10.6 it was pointed out that SCAWIT was making an approximation of the merge because of not having a full properties representation of the meanings. As the by-hand experiment does use a complete properties representation for the meanings, the approximation of the merge is significantly reduced. In addition to the effective disambiguation by having selected a single meaning for each word in, this case a further improvement over the automated version is likely from reducing the approximate value of the combination of the meanings.

10.8 Results

The latest version of SARUMAN is labelled SCAWIT (SARUMAN Clauses Advanced Word Interaction and Topic). Table 10.1 gives the numerical results obtained for the three experiments given in section 10.7 and table 10.2 gives the summary information alongside the key previous baselines.

ID	Human	SCAWIT PoW	SCAWIT Li et al. (2003)	SCAWIT Properties by hand
1	0.78	0.857	0.868	0.910
2	0.91	0.803	0.748	0.939
3	0.27	0.690	0.598	0.607
4	0.44	0.279	0.238	0.557
5	0.24	0.195	0.164	0.291
6	0.07	0.087	0.273	0.09
7	0.18	0.326	0.461	0.561
8	0.11	0.466	0.218	0.100
9	0.37	0.587	0.643	0.700
10	0.37	0.173	0.176	0.580

Table 10.1: Numerical values for SCAWIT using the ten pairs data

Model	Pearson's	Spearman's	R.M.S.
LSA	0.619	0.436	0.211
SARUMAN - PoW	0.207	0.213	0.473
SARUMAN - PoW + typed	0.419	0.395	0.385
SARUMAN - Li et al. + claused	0.637	0.578	0.262
SCAWIT - Li et al. (2003)	0.698	0.650	0.232
SARUMAN - PoW + claused + OPTIC	0.659	0.596	0.280
SCAWIT	0.705	0.584	0.213
Properties – SARUMAN Claused	0.810	0.809	0.207
Properties - SCAWIT	0.872	0.857	0.171

Table 10.2: Summary information for the ten pairs data

The improvement of SCAWIT over SARUMAN with clause disambiguation is at the minimum threshold of what was described as statistically significant as the difference was 0.046 rounding to 0.05 to 2 d.p. With the Li et al. (2003) algorithm, there was slightly above the threshold correlation achieved. Likewise there was an improvement in the by-hand data too, which was giving a statistically significant improvement.

The by-hand properties model was also showing a clear improvement over SCAWIT using the WordNet definitions as was the case with SARUMAN. This is primarily down to the addition of human disambiguated meanings when forming the properties and a refined meaning structure. While the properties model potentially makes the task of automatic disambiguation easier, without constructing a complete database of meanings this cannot be tested. The difference between the SCAWIT and SARUMAN with clauses, however, both had the same advantage from human disambiguation and so suggests that idea of merging clauses gave statistically significant results for the ten pairs dataset.

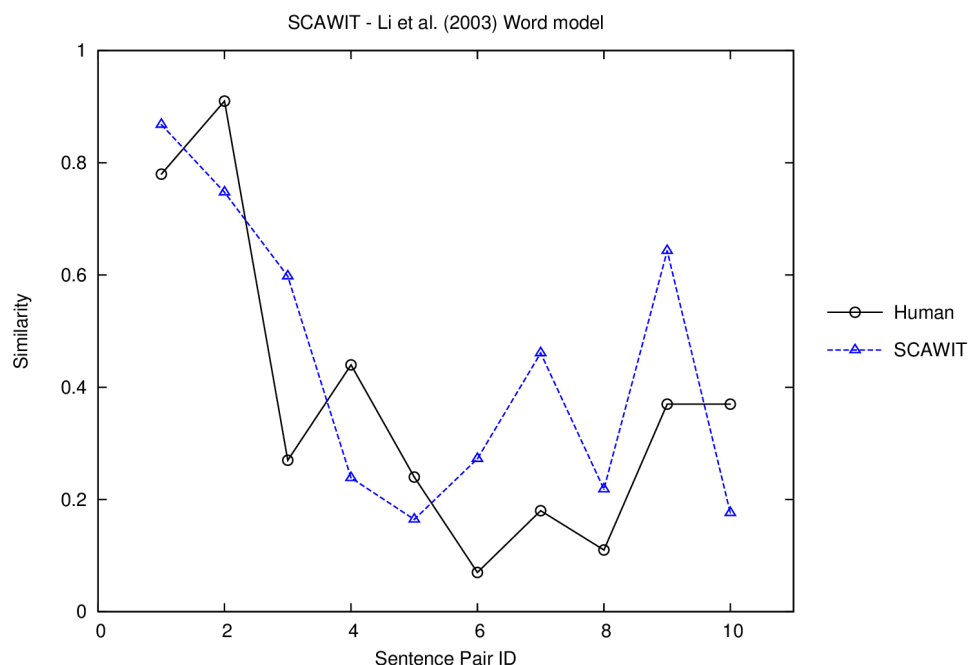


Figure 10.4: SCAWIT – Li et al. (2003)

SCAWIT despite using the conceptually less well suited Li et al. (2003) formula gives a set of values close to the human scores as shown in figure 10.4. Some of the values are perhaps lower than they might be because of the aforementioned limits of word formula underestimating the contribution overlapping meaning. The values can be seen to be visually closer to the human scores than SARUMAN with the Li et al. (2003) formula in 8,5 and with PoW from 9.5.

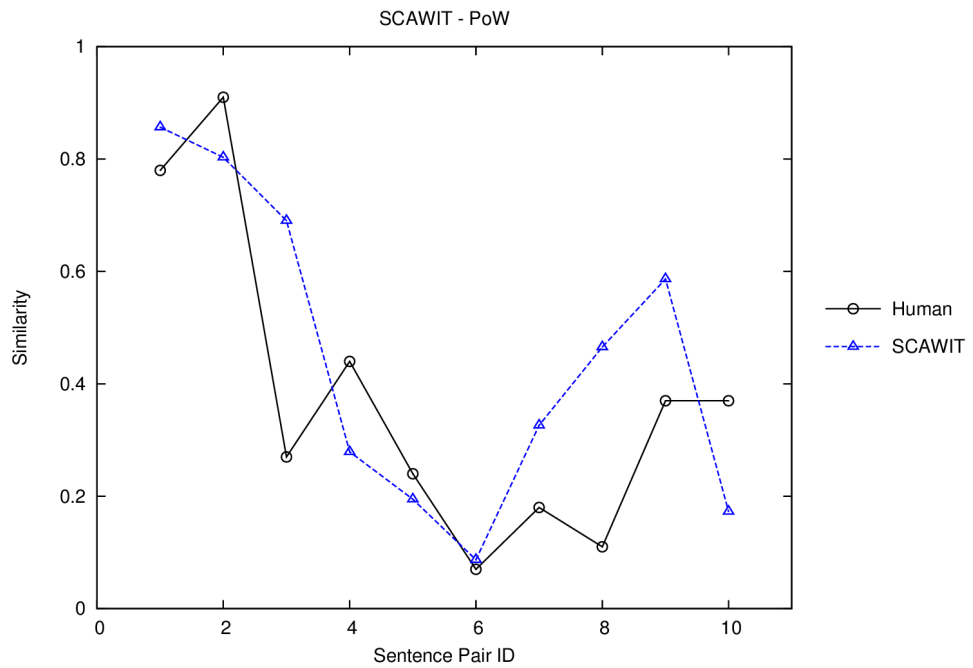


Figure 10.5: SCAWIT (PoW) on the ten pairs dataset

Again as with the implementation with Li et al. (2003), SCAWIT with PoW in figure 10.5 can be seen to a strong resemblance between itself and the human scores. Although pair 8 is further away from the human value than in fig 10.4 the PoW model is less prone to underestimate the similarity.

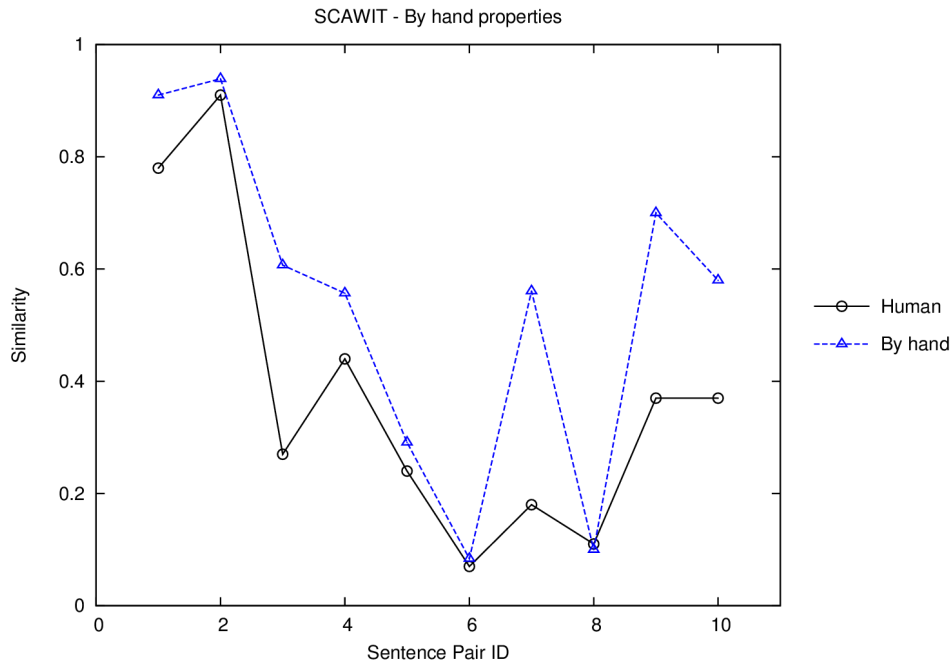


Figure 10.6 Properties by hand with SCAWIT

The constructed properties by hand for SCAWIT in figure 10.6 shows that the inclusion of direct property tagging by a human leads a strong match between the values, although there remains a discrepancy on some pairs.

10.9 Conclusions

This chapter introduced advanced word interaction and the combining of the clauses to SARUMAN to produce SCAWIT. SCAWIT showed statistically significant improvement over the previous version of SARUMAN (0.05 PCF) as defined in the experimental methodology in chapter 4. However, compared to the PoW and OPTIC version of SARUMAN, SCAWIT showed a marginal improvement, but the “by-hand” and the Li et al., (2003) model were showing marginally higher improvement indicating that the combining of words was giving a minor improvement.

These results confirm that the introduction of advanced word interaction and the merging of meanings was an improvement to the sentence similarity model further illustrating that Linguistics is important to sentence similarity, as had been the stated objective of the research.

The properties by-hand version of SCAWIT gave markedly higher correlation than the automated version of SCAWIT. The first reason is that the properties were only constructed for the meanings in the context of the sentence and so removes issues of disambiguation. This is closer to the human tagged meanings from the experiments from chapter 7. With SCAWIT it is also the case that the approximation, to represent how the meanings merge, is greatly reduced from the properties model as this is from where the approximate method was derived .

While a complete representation of the meanings within the knowledge base as properties would still have disambiguation issues, the properties representation would reduce some of the complications with disambiguation that arose with automation using the WordNet definitions. The by-hand results, therefore, only give an indication of higher potential rather than a conclusively better result.

While the by-hand model shows that the method has a higher potential for SCAWIT than has been achieved with the automated version using the primarily WordNet sourced knowledge base. The automated version of SCAWIT is still performing well with a Pearson's correlation of around 0.7.

More importantly, this was the end of the core experimentation of the thesis and while the final improvement might have been small, when compared to the initial mathematical model, it is very clear that the inclusion of Linguistic concepts was both possible and shows clear improvement on the dataset with around a total 0.5 increase in PCF.

The next chapter will examine more closely the reason for the performance of the model and why a better correlation is not being achieved from a Linguistic perspective.

The results signify that the inclusion of Linguistic concepts has improved the mathematical model and yields important improvement for certain types of sentence. The current overall

performance is strong and shows that for some types of sentence that it is outperforming corpus based methods including LSA.

Although, the framework means that SARUMAN cannot be considered a complete work as there are many concepts not yet handled, it is at a stage where it can be assessed for its performance in a wider context. The next chapter examines how it performs with regards to real-time in preparation for testing in specialist domains.

11.0 Discussion and Timings

11.1 Introduction

The last chapter completed the core experimentation to examine whether it was possible to improve a sentence similarity model by gradually incorporating Linguistic concepts. While there are many other Linguistic observations that have not been introduced to the model many are about handling special cases.

The concepts that have been investigated included selecting the right meaning of the word in context, using the functional information in the sentence, the combination of word meanings based upon their Linguistic function and combining meanings where connections exist beyond the functional structure of the sentence.

The core experimentation successfully found that a relative improvement occurred from the addition of the core components to SARUMAN compared to its mathematical version.

In addition to meeting the objective (section 1.4) to show that it was possible to improve a sentence similarity model through the inclusion of implementations of Linguistic concepts, it is also the case that a significant sentence similarity model has been produced. This chapter discusses and analyses the results with respect to the Linguistic features of the ten pairs dataset and where and why SCAWIT is potentially still struggling with matching the human scores.

The other aim of the research given in section 1.4 was to produce a usable model for practical applications that could potentially give greater accuracy than the non-Linguistic models. A key aspect of an algorithm for practical applications is the execution time and to show that at a minimum, that better than real time performance could be achieved. With the exception of chapter 7 where it was decided not to use automatic word sense disambiguation for latter versions, the timing of the model has not been discussed. Section 11.4 gives the timings of a version of SCAWIT which is not fully optimised, showing that it is easily able to exceed the requirement to do a single sentence comparison faster than a

sentence could be typed.

The results have been interpreted simply using the human scores as if they were the actual similarity value. This chapter will include a more in depth discussion to examine where and why SCAWIT is still having issues with the similarity.

11.2 Discussion of core experiment

The core experimentation has been completed and while it has been shown that a clear improvement in the performance of the model occurred from the inclusion of Linguistic concepts, there are still ways in which the model is not matching the results obtained with human tagging.

The interest of the experimentation was to establish whether specific Linguistic concepts could be incorporated into a sentence similarity model and lead to an improvement in the accuracy of the sentence similarity model. However, it is useful to examine more closely the detail of how the algorithms are performing with respect to the ten pairs dataset.

SCAWIT (the final version of SARUMAN) gave a good Pearson's correlation of 0.7 but was still some way short of the similarity achieved of the by-hand properties model using which managed 0.86. The by-hand properties includes human disambiguation in its selection of the meaning of each word prior to selecting the properties and so combined with a more precise combination of meaning overlap and so is beyond SCAWIT's capabilities which doesn't include disambiguation of meaning.

Sentence similarity is essentially an approximate method so it is not likely that it would ever manage to find the absolute similarity of the input sentence pairs, however there is still room for a better performing model on the ten pairs dataset as shown from the human tagging.

One of the significant limitations is in the representation of the meaning. There are places where WordNet (Feldbaum (ed.), 1998) has only given a fraction of the connections

between words. The adjectives "elegant" and "beautiful" have no link between them. The usage of "green" as meaning pro-environment is only contained in WordNet in the loosest sense via "adhering to the policies of the Green Party". The expansion of the vocabulary was aiding the comparison in a couple of places such as matching "has to" and "must".

While it was found that automatic disambiguation using the lemmas of WordNet's definitions was finding some of the intended meanings, it was also found to struggle in many places with the many choices available. However, even were a stronger disambiguation was used, there would still be situations within the dataset where without understanding the meaning, it is unlikely that the right meaning would be selected.

"The car exploded at the art show."

Here, the verb "explode" would normally apply to the usage of a bomb exploding, which is the interpretation when tagged by a person. The problem arises is that there is a higher similarity between an "exploded diagram" and the idea to "show a theory or claim to be baseless" and indeed the verb "shows" in the complementary sentence.

Unless there is enough of a connection between car and explode, maybe via fuel, it would not be possible to select one meaning from the current association and unlikely even with a large corpus. The very large vocabulary including obscure definitions, makes the task of disambiguation very challenging when in an area where there will remain times when any sentence similarity model will struggle.

The automatic parsing of the ten pairs sentences was performing close as to human performance as would be expected from results with other machine parsers. The one case where the tagging was at odds with the human tagging was "the ancient building". As the sentence pair had low similarity, a failure to find the right meaning would make little difference.

Even with a closer matching set of sentences, the selection was made which would still allow a partial meaning to be found via cross-type comparison. So the verb to "build" and the noun "building" would still compare strongly even if fractionally lower than desired.

Similarly, if the structure of the other sentence were the same then it would also be parsed in the same manner, which would still use a like-for-like comparison. So it is clear that the disambiguation by type is performing well.

The other major contribution to the performance on the dataset was finding the clauses. The inclusion of the sentence pair 9, "woman without her man is nothing", highlights the importance of clauses for distinguishing meaning and that of punctuation for determination of the clauses. However, as a sentence pair which is a Linguistic exemplar of the function of punctuation (Taggart and Wines, 2008), it is an exaggerated case of the more common occurrences. Indeed, it was only with the inclusion of clauses that the sentences in pair 9 did not compare with a maximum score with the automated algorithms.

There are a few examples of situations within the dataset where the clauses are merging together: "put their faith in" - "they believed"; "need" - "are essential for"; and "being green" - "environmentally friendly".

11.3 Human scores

So far the human scores have been assumed to closely represent the absolute similarity score but it is worth examining whether these scores are representative of the absolute similarity scores. The one pair which particularly stands out is pair 3:

"Fish swim in water."

"Birds fly in the air."

All versions of SARUMAN are consistently rating this far more strongly with little variation between versions than the mean human rating which scored this sentence pair with a score of below 0.3 (which could be considered as meaning low similarity).

This sentence pair was provisionally given a high rating at 0.8 at the point of creation, although this seems high, another rater in the human raters had also rated this pair highly. The vague connection between the meaning appears to have caused an issue with the

human raters because of the lack of common topic.

There is a common meaning which can be given a common sentence in a similar manner as was done with hypernyms for words:

"Vertebrate animals travel in a medium."

While this is a cumbersome sentence, it is nonetheless a sophisticated common feature which would not normally be possible for a pair of sentences. The model seems to be consistently finding this semantic relationship that was overlooked by most of the participants for the original scoring.

The use of the WordNet ontology and the hypernym relationship made the identification of this common sentence to be found whereas it seems it was not being found by most of the human raters. The connection arises in that the ideas being described are similar but there is almost no overlap of individual words.

One other pair which has a possible issue is sentence pair 9:

"Woman: without her, man is nothing."

"Woman, without her man, is nothing."

While clearly the similarity should be well below unity, there is possibly confusion arising from not just the contradiction but that man and woman are opposites, leading to some raters scoring the similarity as 0. As seen in chapter 14, this is potentially a different scale to the standard similarity scale. However, this underestimate occurring is much smaller than was the case with sentence pair 3.

11.3.1 Pair 3

SARUMAN was consistently giving the same rating for this pair of sentences because of

the common sentence between the two sentences. In contrast the human raters were predominantly not giving a strong connection because none of the terms were directly overlapping each other. In this instance the difference between the automated version and the combined human score can't be attributed to an element lacking from the sentence similarity model.

Conversely, it would appear to be evidence that the human rating for pair 3 could be a weaker approximation of the absolute similarity than was the case for the other sentences due to having a similarity without having overlapping words in both sentences.

Were it to be the case that the sentence similarity model was finding a value of similarity which were closer to the actual desired similarity for this pair of sentences then this could well be affecting the performance of the model on its correlation. The common sentence implies that there was likely an undervaluation by the human raters and it occurred that in the latter replication with the thirty pairs dataset a higher value was obtained. This section briefly examines how the correlations would be affected were a value nearer that being found by SARUMAN and SCAWIT used instead of the human rating for pair 3. This then gives an indication of the possible impact of the noise from the human ratings may be have.

Table 11.1 shows the correlations were the score for sentence pair 3, in the ten pairs dataset, be adjusted to a value of 0.7 which is much closer to the model's score than the human scores. While it is the case that the adjusted value could be closer to the absolute similarity, as indicated by the results, there is the caveat that this does not guarantee that it is the only obtained value which is significantly different from the absolute semantic similarity. There have been no other such results highlighted by the model. Nonetheless, it is likely that the adjusted ten pairs dataset is closer to the absolute similarity and therefore with lower noise than the raw data.

It can be seen that the correlation of the model increases for almost all instances of SARUMAN but the pure mathematical model. The relative improvements between the values of the model remain and the difference between SCAWIT and clause disambiguation increases fractionally to just over the 0.05 without rounding.

Model	Pearson's	Spearman's	R.M.S
LSA	0.405	0.247	0.350
SARUMAN - PoW	0.166	0.225	0.449
SARUMAN - type disambiguation	0.485	0.492	0.353
SARUMAN - claused + OPTIC	0.762	0.742	0.237
SCAWIT	0.814	0.717	0.167
By hand properties - claused	0.835	0.857	0.180
By hand properties - SCAWIT	0.867	0.869	0.185

Table 11.1: The summary information for ten pair dataset with an adjusted pair 3 value of 0.7.

It was likely that moving a single value closer to the value obtained by SARUMAN would improve the correlations this is not guaranteed for any metric other than the RMS that an improvement would occur. The fact that the improvement between versions still shows statistical significance confirms that the earlier results were very unlikely, due to noise in the human ratings.

The difference between the by-hand properties representations, however, narrow slightly which show that the difference between the two versions is small for the ten pairs dataset.

If the changed value for the dataset is used the Pearson's correlation of LSA falls significantly to 0.41 which then would clearly be performing significantly worse than SCAWIT which is over 0.4 higher. It would not be surprising that LSA is struggling with some of the concepts contained in the dataset discussed in the previous section, as it was already mentioned as a probable short-coming that LSA ignored word order and punctuation in the literature (Li et al., 2006).

While for consistency the human ratings should be used for the benchmark without consideration as to what the actual similarities are, the results with the adjusted score do indicate that the noise on the system was hindering, rather than helping the correlations for the ten pairs dataset.

11.4 Timings

One important area which has not yet been discussed, is the performance of the algorithm with respect to the time taken to run. It was earlier stated that the ability to process a pair of sentences in real-time was important and was a factor in the decision not to continue with the automatic meaning disambiguation (chapter 7) for part of the continued experimentation.

While the sequential parser (section 8.2) is an optimal approach, the code is not fully optimised. Therefore the purpose of these timings is simply to demonstrate that real-time sentence similarity is possible with the different versions of SARUMAN. Real time means that a pair of sentences could be compared in the time taken to type a single sentence, which is the requirement to enable parallel processing.

The timings are given for the ten pair dataset and the test dataset for the (Microsoft Research Paraphrase dataset) MSRP, (Dolan et al., 2004) used for the next chapter. Although, the MSRP consists of much longer sentences, most of the difference between the two algorithms without parsing is due to pre-loading the entire sentence set and tagging from the parser which then leads to memory issues. The order of increasing complexity with size should be nearer order 3 for optimised code.

The times are given after initialisation which pre-loads all of the vocabulary into a standard structure such as a map and takes a couple of minutes. A rate of around 100 sentence pairs per minute should be considered better than real-time.

The system specifications for the timings were: Microsoft Visual C++ on Windows XP with a 3GHz processor with 3GB of RAM and the results can be found in table 11.2.

Task	Dataset	Repetition	Sentence pairs (millions)	Time (minutes)	Time for a million pairs
Parsing	MSRP	1000	1.725	40 min 55 sec	23 min 43 sec
Parsing	Ten pairs	100000	1.000	8 min 1 sec	8 min 1 sec
SARUMAN (No parsing)	MSRP	1000	1.725	50 min 58 sec	29 min 33 sec
SCAWIT (No parsing)	MSRP	1000	1.725	63 min 36 sec	36 min 52 sec
SARUMAN (No parsing)	Ten pairs	10 million	100.0	14 min 54 sec	8.94 seconds
SCAWIT (No parsing)	Ten pairs	10 million	100.0	15 min 8 sec	9.08 seconds

Table 11.2 Timings for SARUMAN (with clause disambiguation) and SCAWIT

Both models (SARUMAN and SCAWIT) manage considerably better than real-time execution even when including the parsing and although some critical algorithms are fully optimal, there are many suboptimal components. For the ten pair dataset, the parsing dominates the processing time and SCAWIT can manage a rate of 176 million sentence pairs a day or 2040 pairs per second. The MSRP results in a rate of 23.8 million sentence pairs a day or 275 pairs per second. Sentence similarity is perfectly parallelisable provided there is the time needed to process a single sentence pair.

11.4 Conclusions

The Linguistic concepts that have been added to SARUMAN at this point, since its mathematical version in chapter 6, are very general concepts which will be found in almost all sentences and fragments and are relevant to determining their similarity.

However, the impact of including a specific Linguistic idea to a sentence similarity model will depend upon the input dataset. This is because the relative change depends, not only upon the presence in an input sentence, but also in the variation between the datasets. This means, although there would be an expected increase in accuracy over a dataset from using disambiguation, clauses and combining meanings, that the magnitude of this effect is unknown.

What is indicated is that for any arbitrary pair of sentences that the output will have a high probability of being more accurate through the inclusion of these Linguistic ideas. Therefore, it is reasonable to view such a model as being superior.

Although, there are many Linguistic ideas that apply to certain types of sentence, these are of a much lower frequency and too numerous to contain in a general or comprehensive scored dataset. The model is showing excellent processing time compared to the requirement for real-time comparisons.

While it can be seen that there are ways in which SCAWIT has not matched human understanding, the results suggest that it is at a level where it would be useful as part of an automated system. The results show solid correlation to human understanding and the ability to work much faster than a human.

Following on from the discussions on the ten pair dataset, the next chapter will use an expanded version of the dataset to form a better benchmark for when algorithms have very different architectures such as LSA and SCAWIT.

12.0 Benchmark: Thirty Pairs Dataset

12.1 Introduction

The core experiment had been able to use the same small dataset because most of the parameters between versions were constant, significantly reducing the noise on the experiments. The aim of showing that including Linguistic concepts could potentially produce a more accurate sentence similarity model than those not following a Linguistic approach, was indicated with the properties by-hand version of SCAWIT.

However, the size of the ten pairs dataset is too small when comparing sentence similarity models adopting very different approaches because of the potential noise that could result from the knowledge base. SCAWIT showed a significantly higher value for the Pearson's correlation for the ten pairs dataset than LSA, The discussion in the previous chapter suggests that the improvement is not an anomaly with the noise from the human judgement, possibly favouring LSA over the SARUMAN algorithm.

This chapter continues with treating the implementation of SCAWIT as a stand alone computer model and performs an experiment with an expanded version of the ten pairs dataset first detailed in section 3.6.

The use of this expanded thirty pairs dataset is threefold. Firstly, to provide validation that the performance of SCAWIT is repeated on the larger dataset, confirming that the earlier results for the ten pairs dataset were not purely an artefact of the dataset but that similar results would be returned for the larger dataset. Secondly, confirmation that SCAWIT is still outperforming LSA, which had proven the most resilient of the pre-existing models tested with the new dataset, on a larger dataset. Thirdly, to provide a benchmark that could be used to compare future models against SCAWIT in the future.

The framework is the core component to the ability to have gradually extended SARUMAN and showing that the inclusion of Linguistics could improve a sentence

similarity model, However, SCAWIT does more than just provide evidence that Linguistics can improve a sentence similarity model but also is being considered as a complete functioning sentence similarity model as it now stands even though there has been identified potential improvements.

The benchmark provides the option for future models to be compared against SCAWIT with its current knowledge base without the need for re-implementation or access to the original code. While a single dataset can only give an indication of the relative performance of models with different knowledge bases, it can be useful to have this indication.

The thirty pairs dataset is large enough to give a signal of the similarity that would dominate the potential noise. While it is not supposed to represent language as a whole it does provide a dataset containing basic Linguistic variation lacking from the standard dataset in the literature as discussed in chapter 3.

This gives a benchmark that people could aspire to beat in the future, however, it is not the limit of the approach of the framework and SARUMAN as shown by the properties of words by hand versions and there is still scope for further evolution of the model such as will happen for the specialist area of opposites in chapter 14.

12.2 Results

The numerical values obtained for both LSA (LSA: on_line implementation, 2013) and SCAWIT can be seen in table 12.1. The summary information is given in table 12.2 and it can be seen that SCAWIT outperforms LSA on all metrics. This is reflected in the graphical representation where it can be seen that LSA is noisier than SCAWIT.

The Pearson's correlation function shows that the results for SCAWIT are consistent with those obtained for the smaller ten pairs dataset. While the obtained 0.69 leaves scope for superior performance nevertheless this represents a strong performance.

ID	Sentence Pair	Human	LSA	SCAWIT
1	<i>The red car was illegally parked on the yellow line . The cake was eaten by the hungry boy.</i>	0.01	-0.05	0.14
2	<i>The glass of water is on the table. The book was atop the dresser</i>	0.29	0.03	0.53
3	<i>I heard the birds singing in the morning. I like listening to birdsong.</i>	0.41	0.34	0.77
4	<i>The fast had lasted all day. The car was speeding for the whole journey.</i>	0.09	0.08	0.09
5	<i>The man who was standing by the river is the president of the company. My boss is standing beside the river.</i>	0.56	0.47	0.82
6	<i>The acrobats and tumblers were my favourite. I now need glasses to read my favourite book.</i>	0.16	0.48	0.54
7	<i>He shot the rifle at the rabbit. The woman photographed the giraffe.</i>	0.33	-0.05	0.55
8	<i>The boat floats on the surface of the water. A hovercraft glides on a cushion of air.</i>	0.60	0.1	0.72
9	<i>Butterflies that flourish on grassland across Europe are in steep decline, indicating a catastrophic loss of flower rich meadows in many European countries. Wild populations of bumblebees appear to be in significant decline across Europe.</i>	0.48	0.64	0.51
10	<i>It was not the sales manager who hit the bottle that day, but the office worker with the serious drinking problem. That day the office manager, who was drinking, hit the problem sales worker with the bottle, but it was not serious.</i>	0.35	1.0	0.41

Table 12.1a: Pairs 1-10 of thirty pair dataset for LSA and SCAWIT.

ID	Sentence Pair	Human	LSA	SCAWIT
11	<i>The man hammered the brass hook into the wall. He screwed in the shiny screw.</i>	0.45	0.17	0.73
12	<i>The ice cracked beneath their feet The leaves rustled in the wind.</i>	0.14	-0.02	0.42
13	<i>Water has been found on Mars! Water was found on the bathroom floor.</i>	0.60	0.64	0.80
14	<i>The rocket launched the satellite into orbit. The cement had held the bricks together for over a century.</i>	0.04	0.03	0.29
15	<i>Estate agents sell houses and flats. Greengrocers trade In fruit an vegetables.</i>	0.50	0.11	0.73
16	<i>The car was destroyed by a tree. The falling branch crumpled the automobile.</i>	0.75	0.53	0.83
17	<i>A passer-by was killed by a knife wielding maniac. The maniac stabbed a passer-by who died in hospital.</i>	0.85	0.30	0.55
18	<i>The barman had diluted the drinks. The owner of the pub had added water to the beer.</i>	0.81	0.34	0.45
19	<i>The sound of the violin brought tears to the audience. Music can sometimes make me cry.</i>	0.40	0.22	0.73
20	<i>The box was too small for the book to fit in. The men were fighting over the ticket.</i>	0.03	0.02	0.28

Table 12.1b: Pairs 11-20 of thirty pair dataset for LSA and SCAWIT.

ID	Sentence Pair	Human	LSA	SCAWIT
21	<i>The Persian cat sat on the carpet. The ginger cat sat on the mat.</i>	0.83	0.91	0.86
22	<i>The caterpillar metamorphosed into an elegant butterfly. The caterpillar changed into a beautiful butterfly.</i>	0.90	0.73	0.80
23	<i>Fish swim in water. Birds fly in the air.</i>	0.56	-0.02	0.69
24	<i>They believed the red bus was environmentally friendly. They put their faith in the train being green.</i>	0.45	0.30	0.28
25	<i>To drive a manual car, you must press down the clutch. To open the window, the mouse has to be double clicked.</i>	0.30	0.05	0.19
26	<i>The green grass glimmered as the sun shone on the morning dew. The ancient building had stood on that small hill for eons.</i>	0.03	0.08	0.09
27	<i>The Persian cat sat on the carpet. The Persian rug was on the dresser.</i>	0.27	0.43	0.33
28	<i>The exploded diagram shows how cars work. The car exploded at the art show.</i>	0.07	0.35	0.47
29	<i>Woman, without her man, is nothing. Woman: without her, man is nothing.</i>	0.40	1.00	0.59
30	<i>Trees need sunlight and water to grow. Food and drink are essential for your development.</i>	0.39	0.08	0.17

Table 12.1c: Pairs 21-30 of thirty pair dataset for LSA and SCAWIT.

Model	Pearson's	Superman's	RMS
LSA	0.465	0.513	0.311
SCAWIT	0.693	0.735	0.224

Table 12.2: Summary information for LSA and SCAWIT on the Thirty Pairs dataset.

Since sentence similarity is an inherently approximate approach to handling the meanings of pairs of sentences, it is not surprising that there are some sentence pairs where SCAWIT does not give a particularly close answer to the human response, even without clear issues of failure to disambiguate.

The dataset is computationally challenging. This is perhaps highlighted best with the sentence "Water has been found on Mars". Not only are there disambiguation issues for the meaning of "water" but there is also wider meaning and implication of the sentence as a whole, because the statement is significant with other facts that readers might be familiar with as regards the planet Mars. The presence of water on Mars, of course, being highly significant to the viability of life, explaining why the human similarity scores are significant lower for that pair than the output of the SCAWIT model.

There were, as to be expected, disambiguation issues on the vocabulary (discussed in chapter 7). Sentence pair 6 has an issue with "glasses" and "tumblers" both taking a different meaning in context, than their highest overlapping possible meanings as in: "containers for drinks".

Limitations in the knowledge database and meaning representation add additional limitations to the overall similarity accuracy. There are some examples where SCAWIT is overestimating the similarity from the underlying similarity of the words and structure beyond trivial issues of disambiguation such as in "rustled" and "cracked". In this case the connection between the verbs from the underlying meaning structure gives too high a similarity for the given context, even with potentially the correct meanings selected.

There are also situations where SCAWIT is underestimating the meaning because it is not

finding some of the more complex connections between meanings. So that "knife-wielding" and "killed" with "stabbed" are not being connected. However, several connections of non-trivial relationships in the meanings are being found.

Even with the potential for further enhancement, as discussed in chapter 11, as a whole the model is performing solidly as reflected by the Pearson's correlation.

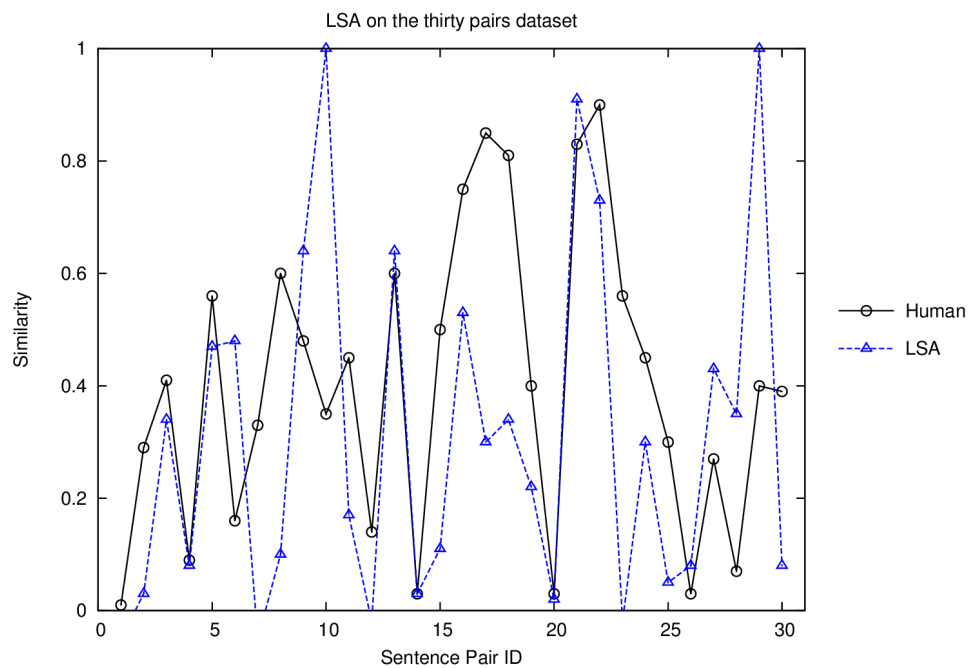


Figure 12.1 LSA for thirty pairs dataset

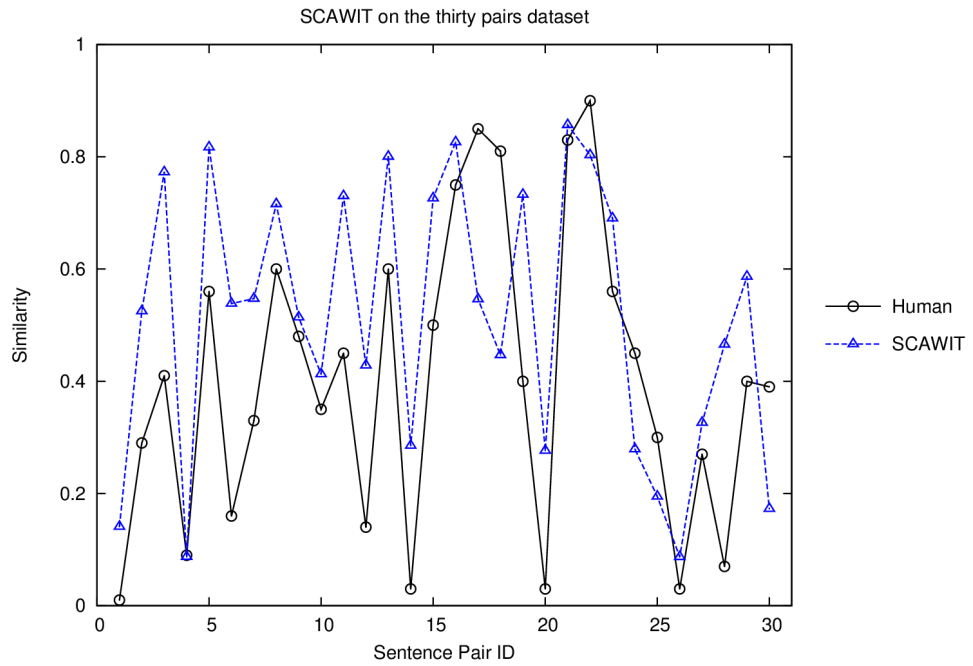


Figure 12.2 SCAWIT on the thirty pairs dataset

The graphs for SCAWIT (figure 12.2) and LSA (figure 12.1) reflect the summary metrics. LSA shows significant trouble in matching the human scores and while the last ten pairs are the same as for the previous experiment, the failure to match is more pronounced. Pair 10 represents the same failure as pair 12 but there are many examples where LSA is failing to find the similarity of the sentences and others where it is over estimating.

While there are sections where SCAWIT is overestimating the similarity (i.e. runs 5, 6 & 7) and a few underestimates, on the whole the performance is strong and the meanings are finding connections that LSA is failing too. While there are several regions where SCAWIT is not matching the human scores, it is much more consistent with the human scores than LSA.

12.3 Conclusions

There is a very clear improvement in the Pearson's correlation for SCAWIT compared to LSA, over a 0.22 increase, confirming the earlier result that SCAWIT is outperforming LSA in several key respects.

As LSA is a corpus method that was performing solidly on the earlier tests, it strongly supports the case that it is the inclusion of the Linguistic concepts that is allowing SCAWIT to outperform LSA.

The dataset is large enough to potentially provide a benchmark for future sentence similarity models to be compared upon, even if having a fundamentally different approach or knowledge base to SCAWIT. Importantly, it contains sufficient Linguistic variation to better represent some of the general issues that can commonly occur in English. Although, there will be specialist situations which are not included in the dataset which could require a specialist dataset.

The next chapter examines SCAWIT on a specialist domain of paraphrases.

13.0 Paraphrase Domain

13.1 Introduction

This chapter is not part of meeting the objective of the research but presents a set of experiments to further evaluate SCAWIT as a complete stand alone functioning model. Chapter 13 examines applying the sentence similarity model to a specialist domain,

This chapter examines the performance of the semantic sentence similarity model on a domain specific task of identifying paraphrases. Two versions of SARUMAN are used for this domain. The mathematical model from chapter 6 and the most advanced version using Linguistics, SCAWIT, from chapter 10.

A paraphrase is not a description of the similarity between a pair of sentences but a particular relationship between a pair of sentences. Which means that both sentences have the same meaning. Therefore, this results in paraphrases having a high semantic similarity to each other by definition. Paraphrases are not directly equivalent to the similarity of a sentence because paraphrases can be dependent on whether a pair of sentences contain a contradiction or not, regardless of the impact of the contradiction.

The method and adaptation of the sentence similarity model for use with the MSRP, Microsoft Research Paraphrase dataset (Dolan et al., 2004) discussed earlier in section 3.4, is given in sections 13.2 and 13.3.

Section 13.4 presents the results from SARUMAN and SCAWIT alongside the reported results from other models in the literature. The MSRP is the largest extant set of human classified sentence pairs and an in-depth evaluation of the test dataset is given in section 13.5. As will be shown later using a sentence similarity model for the paraphrases in the MSRP raised some issues for SCAWIT for the task and these are also examined.

13.2 Method

This experiment will use the (MSRP) Microsoft research paraphrase dataset which has been randomly divided into a test set of 1725 sentence pairs and the training set of 4076 sentence pairs (Dolan et al., 2004).

The set of sentences are generated from a common source which means that all of the sentence pairs will have a high or very high similarity as a pre-requisite. So even when the sentences are determined as being non-paraphrases, the sentence similarity model should return a high similarity score.

The associated press releases a news story which is then re-reported throughout numerous sources, often being rewritten, but conveying the same information. Since the article as a whole should be conveying the same information as the original source, this means that the sentences will be very close to paraphrases even where judged not to be paraphrases to one another. This means that the dataset as a whole can also be regarded as distinct from a random sentence pair. An examination of the mean level of similarity scores will be used to judge the ability to distinguish the dataset as a whole from random data, although there is no clear benchmark to use for this judgement.

Both SARUMAN and SCAWIT were run on the MSRP using the training set to determine a simple threshold above which all sentence will be classified as a paraphrase and the results for the test set compared against the other results reported in the literature. The threshold was restricted to 0.6 and above using steps of 0.01.

13.3 Mean Similarity and Threshold

The threshold in similarity to declare a pair of sentences a paraphrase for SARUMAN was 0.62. Which is similar to the threshold declared in the literature for STS / IISIS (Islam and Inkpen, 2007) of 0.6 (using 1 d.p.) and 0.55 for SSA (Hassan, 2011) and higher than

CHESA (Lieberman and Markovitch, 2010) which used 0.3. For SCAWIT though, a significantly higher threshold was obtained of greater than or equal to 0.8.

From the initial definition of high similarity (chapter 3), a value of above 0.7 was required and so for consistency, it would be expected that a pair of paraphrases would score at least this level of similarity. Thus, in that respect, SCAWIT is behaving as anticipated and closer to the expected scale than the other reported thresholds. The higher level is conceptually more consistent with the definition of a pair of paraphrases being highly similar.

When examining the mean of the similarity scores for the entire dataset, SARUMAN gives 0.732 and SCAWIT almost 0.1 higher at 0.831. This would seem consistent with the close meaning that the non-paraphrases in the dataset have which would also be expected to score strongly in similarity. Therefore, it would be reasonable to view SCAWIT as probably better than SARUMAN at distinguishing the dataset (so near paraphrases and paraphrases) as whole from a random sentence pair. However, this conclusion is in part speculative and for other models is not a metric which has been recorded in the literature.

13.4 Paraphrase Identification

The more important evaluation is how the sentence similarity model is performing for the task of identifying the pairs classified as paraphrases in the dataset. As seen by the literature the MSRP has been processed by a large number of models: semantic sentence similarity; general relatedness; and specialist paraphrase identifiers. Their results and those for SARUMAN and SCAWIT can be seen in tables 13.1 and 13.2.

Model	Accuracy	f-measure	Precision	Recall
SARUMAN	0.711	0.802	0.736	0.881
SCAWIT	0.714	0.797	0.753	0.847
<i>From Mihalcea et. al (2006)</i>				
Random	0.513	0.578	0.683	0.50
VBS (baseline)	0.654	0.753	0.654	0.716
<i>Knowledge Based</i>				
Wu & Palmer (1994)	0.690	0.800	0.720	0.921
Resnik (1995)	0.690	0.804	0.690	0.964
Lesk (1986)	0.693	0.789	0.724	0.871
Jiang & Corinth (1997)	0.693	0.789	0.724	0.866
Lin (1998)	0.693	0.790	0.716	0.887
Leacock and Chodorow (1998)	0.695	0.790	0.724	0.870
Compound (<i>Mihalcea et. al., 2006</i>)	0.703	0.813		
STASIS from Hassan (2011)	0.668	0.799	0.673	0.983
<i>Corpus Based</i>				
<i>From Hassan (2011)</i>				
LSA	0.684	0.805	0.697	0.952
ESA	0.688	0.799	0.700	0.929
PMI-IR	0.699	0.810	0.702	0.952
SSA	0.725	0.814	0.739	0.907
STS (IISIS)	0.726	0.813	0.747	0.891

Table 13.1: Similarity models paraphrase identification scores for MSRP test set

Model	Accuracy	f-measure	Precision	Recall
CHESA (Lieberman & Markovitch, 2010)	-	0.797		
Rus et al. (2008)	0.706	0.805		
Zhang & Patrick (2005)	0.719	0.807	0.743	0.882
Qiu et al. (2008)	0.720	0.813		
Lintean et al. (2010)	0.726	0.810	0.740	0.893
Mitchell & Lapata (2010)	0.730	0.823		
Blacoe & Lapata (2012)	0.735	0.822		
Fernando & Stevenson (2008)	0.741	0.824		
Ul-Qayyum & Altaf (2012)	0.747	0.818	0.782	0.8658
Finch et al. (2005)	0.750	0.827		
Wan et al. (2006)	0.756	0.830		
Das & Smith (2009)	0.761	0.827		
Kozareva & Montoyo (2006)	0.766	0.796	0.944	0.688
Socher et al. (2011)	0.768	0.836		
Madnani et al. (2012)	0.774	0.841		

Table 13.2: Supervised specialist paraphrase identification models and others.

The two summary metrics of classification accuracy and f-measure, show that SARUMAN and SCAWIT are giving similar performance to one another with SCAWIT giving a marginally higher accuracy with 0.714 as compared to 0.711 which is an improved classification of 5 sentence pairs. However, the f-measure which is important as regards information retrieval, showed the reverse with SCAWIT giving 0.797 and SARUMAN (using PoW) 0.802. While the difference is small, for such a large dataset this could potentially be significant. Looking at the precision and recall, it can be seen that SCAWIT has a significantly higher precision than the earlier version of SARUMAN but does so at the expense of failing to identify several of the paraphrases. With the much higher threshold of similarity, this change in the relative values of the precision to the recall are unsurprising.

Mihalcea et al. (2006) provided two benchmarks of vector based similarity and a random

algorithm. It can be seen that a definite superior performance is achieved over the benchmarks. This shows that the knowledge in SARUMAN is helping in identify the paraphrase and therefore could help assist as part of automation. However, the same can be said for almost all the other models tested on the dataset.

In comparison to the knowledge based word algorithms reported in Mihalcea et al. (2006), it can be seen that SCAWIT does slightly better than the other knowledge based approaches, but only comparable for the accuracy and worse for the f-measure for the supervised combination of all the other algorithms.

When compared against the sentence similarity models already being assessed in chapter 6, it can be seen that that SCAWIT does better for accuracy than LSA and STASIS but slightly worse for the f-measure. However, when compared against IISIS (STS) it can be seen that a definite increase is managed by IISIS over SCAWIT for both metrics. Similar results were obtained for another corpus method: SSA (Hassan, 2011) but two further corpus methods are weaker than SCAWIT (ESA and PMI-IR).

It can be seen that there are several models which are showing clearly better performance at the task of identifying the paraphrases from the MSRP corpus than SCAWIT. Especially, amongst the models using specialists algorithms combined with supervised learning for multidimensional factors.

13.4.1 Specialist Methods

There were several methods that were specifically designed and trained for the task of paraphrase detection. The strongest performing from amongst the specialist models is Madnani et al. (2012) which uses Machine Translation algorithms of several, already strongly performing (0.743 accuracy) models, in some cases especially designed for paraphrase identification such as TERp (Snover et al., 2009). The introduction of the other 7 metrics introduces issues of greater noise relative to non-compound metrics from the risk of over-fitting and it is not clear that it is statistically significantly better than algorithms which match the 0.756 accuracy that Madnani et al. (2012) reported for their model when

just including TERp.

The most significant approach, displayed by the other strongly performing models, is using the matrix of word meanings and then using this with supervised training on the 4076 training pairs of sentences. Unlike the methods which reduced the matrix to a single value, the matrix allows for different levels of similarity to be applied to different situations. The most successful, Socher et al. (2011), includes information from a parse tree as was the case for machine translation.

One other approach to paraphrases of particular interest was the use of dissimilarity as employed by Qiu et al. (2008) because the contradictions, which can be used to define when highly similar sentences differ, remain constant even when an extra similar component is added to each pair. This approach did not show a big improvement over the similarity metrics though.

13.4.2 SCAWIT

While it could be said that both SARUMAN and SCAWIT are performing solidly for the task of paraphrase identification, from the results for the core experimentation, it would have been expected that SCAWIT would show a much better performance than it has done relative to the non specialist algorithms. The most notable feature is that SCAWIT failed to surpass the performance of the raw form of SARUMAN using PoW. With the dramatic difference that was seen between the versions of SARUMAN when incrementally tested on the ten pair dataset, it would have been expected to see a clear improvement in performance, instead the f-measure is marginally worse.

13.5 Errors in the MSRP dataset

As a result of weaker than anticipated performance of SCAWIT, the results were examined pair by pair. The first thing to note is that there is substantial error rate within the human ratings within the dataset. It has been noted that there was inconsistency between the

guideline handling and the raters interpretation (Madnani et al., 2012) but there also remains clear internal inconsistency.

There are some clear cut examples of sentence pairs which cannot be accurately classed as paraphrases (particularly notable in financial news data):

"The last time the S&P had a larger one-day point loss was also May 19, when it gave back 23.53 to close at 920.77."

"The last time it had a larger one-day loss was July 1, 2002, when it shed 59.41 to close at 1,403.80."

There is a clear contradiction between even the main clause with the same event supposedly happening on different dates.

It is not as simple to declare a pair of sentences classed as non-paraphrase as definitely a paraphrase but perhaps the following from the test set:

"NBC will probably end the season as the second most popular network behind CBS, although it's first among the key 18-to-49-year-old demographic."

"NBC will probably end the season as the second most-popular network behind CBS, which is first among the key 18-to-49-year-old demographic."

this would be an example of how close the meanings can be. The difference between "although" and "which" could be argued to be enough to make them non-paraphrases but this is not logically clear and in the context of other scores, would seem inconsistent.

In fact this pair makes it very clear how inconsistent the human ratings can be, as just two pairs earlier in the test dataset the same pair of sentences in reverse order of presentation were scored as paraphrases.

There are well over 100 pairs which should be judged as wrongly classified in the test set with it being around 10% error rate or slightly higher. However, due to subjectivity, in many cases a definitive declaration cannot be made as to where precisely the logical line

should come but nonetheless the overlap is marked between inconsistently rating pairs.

For another illustration the first pair with larger superfluous information than for the second is rated as a pair of paraphrases and the latter not:

"Last year, he made an unsuccessful bid for the Democratic nomination for governor."

"He ran last year for the Democratic nomination for Texas governor, but lost the primary to multimillionaire Tony Sanchez."

"Shares ended Wednesday at \$6.83, up 2 cents."

"Shares of Goodyear rose 2 cents on Wednesday and closed at \$6.83."

It seems likely that in some cases that the discrepancy arises from inconsistency due to lack of concentration which is unsurprising when you consider that there are over 200,000 words in the dataset to read even prior to rating the pairs. So the first example error given could have resulted either from wrong data entry or simply not reading beyond the first couple of words.

Although, there is some accompanying analysis of how the raters corresponded with each other for their initial rating (Dolan, 2005), there is insufficient data to make any meaningful assessment of the ratings due to no information on when the deciding third rater was needed.

If assuming equal probability of being right for every rating then the original ~83% correspondence would indicate about 97.4% reliability when considering the third rater's involvement. Conversely, if assuming that for any situation there is either a clear answer or the rater makes a blind guess, then the sure answers would be at 66% with a possible overall accuracy of slightly below 83%. In practice, there will be a range of complexity and chance of a rater being right which would make the 10% error rate seem plausible.

When looking at the original estimates of rater 1, (who assessed all of the pairs) who gave 62% and the others over 72% paraphrases with the final total being ~67% is also consistent in suggesting a 10% error

Several of the misclassifications are clearly picked up by the algorithm, for example SCAWIT rated the following from the training set as a Paraphrase:

"The ECB has cut interest rates six times over that period, from 4.75 percent in October 2000 to 2.5 percent."

"The ECB has cut rates from 4.75 percent in October 2000 to 2.5 percent in that period."

However, while this would suggest that the error bars on the accuracy rating for a model on the MSRP is large, at around +/- 5%, it will be the case that most of the erroneous sentence pairs which are easily picked up will result in the same success or failure for all of the models. So the relative performance of each model will have much smaller error bars in most cases until approaching the upper limit of performance, which would be higher than is currently being managed.

13.6 Issues with SCAWIT

Although no obvious weakness accounting for SCAWIT's performance in identifying paraphrases from an error could be identified, there were some clear situations where it was struggling with giving the correct similarity.

Firstly, there were a large number of nouns which were outside of WordNet's vocabulary and on occasion the proper nouns were overlapping real words. This meant that a name such as "Mike Butcher" could lead to obscure valid parsing. As a result there were a substantial number of wrongly parsed sentences, although not all of these lead to a failure to correctly classify the sentence pair.

The second issue was when there were long compound sentences. There were sentences which had 5 main verbs and the combination of multiple main clauses by SCAWIT would often give weaker similarities than expected. Similar situations arose with some quoted statements where the main verb was "said" or a synonym.

These although significant, were not the main cause of the under performance of SCAWIT as a paraphrase identifier but rather address more general issues for further investigation for improving the parser and compound sentences with conjunctions.

13.7 Saturation of Similarity

The major issue which arises between SCAWIT and SARUMAN is that in many cases the improvement in similarity is resulting in failure to distinguish paraphrases. It was already stated at the beginning of this chapter that paraphrases were not the same as similarity and the closeness in meaning of the sentence pairs inside the MSRP means that similarity is not yielding enough resolution for distinguishing the pairs.

With a large amount of unhandled information, such as proper nouns and numbers, not in WordNet and numbers, the increase in vocabulary for SCAWIT is strengthening the similarity and lessening the impact of differing words between the sentences. For simpler models, it was found that including more than n-gram matching was an advantage (Islam and Inkpen, 2008) but there comes a point where a contradiction is lost.

The similarity between words is already approximate but it is not a lack of precision in the output of the sentence similarity model which is the issue. Within a subset of a particular group of sentences, there might well remain a consistent change in similarity, although smaller than before but between the subsets a cross-over in similarity occurs.

So the if the mean rises from 0.7 to 0.8 for one subset but 0.73 to 0.77 for another then the result is that the distinction between the subset reduces. There are cases where there is saturation of similarity and no matter how much the similarity is improved, a threshold is not going to allow a highly similar non-paraphrase to drop beneath the threshold, while recalling enough of the lower similarity pairs.

"A tropical storm rapidly developed in the Gulf of Mexico Sunday and was expected to hit somewhere along the Texas or Louisiana coasts by Monday night."

"A tropical storm rapidly developed in the Gulf of Mexico on Sunday and could have hurricane-force winds when it hits land somewhere along the Louisiana coast Monday night."

For example other than the tense, the core distinction between this pair of sentences is that the first case is "Texas or Louisiana coasts" and in the second only "Louisiana coast", it is the latter contradiction which is not reflected in the similarity.

13.8 Future Work

While there are two or three issues that have been highlighted in the SCAWIT algorithm which could be implemented to improve the similarity score and its performance on identifying paraphrases, a preliminary investigation strongly suggested that these improvements would still be insufficient to overtake the best specialist algorithms for identifying paraphrases.

A paraphrase is basically determined from identifying contradictions and the level of similarity to dissimilarity. To find these differences, the two sentences want to be aligned with one another, accounting for the amount of overlap, and then processing the remaining sections.

Linguistics can help identify when clauses align as near paraphrases when using different words and significantly different order. Then the type of relationship between the difference, such as adverbial temporal clause (i.e. "On Wednesday"), can be used to decide on a contradiction or merely extraneous information.

To do this requires comparing two chains "A B C D " to "A b D E F c" which is easy to describe to a human but complex to accurately implement by automation. Nonetheless, a combination of clauses, similarity and relatedness judgement with the leading existing approaches could have the potential to go significantly beyond that which is currently managed on the field.

However, this is very different from the sentence similarity algorithm being developed and is well outside the scope of the research presented in this thesis and would require parsing the sentences in tandem and a new parser.

Increasing the vocabulary using a properties model could also, as noted earlier, refine the similarity comparisons distinguishing the similarities which are close with a better resolution.

The issue with compound sentences is not clear as to whether it is one which should be regarded as part of the sentence similarity or one of document processing and general questions over similarity. A question arises of how to compare A & B against A. This could be the case that B is regarded as all distinct properties or it could be compared to A too. With sentences there is a question as to whether the sentence should be split prior to even presenting the sentences to a model.

"The cat sat on the mat and the dog went for a walk."

"The cat sat on the mat. The dog went for a walk."

It could be the case that the first utterance should not be compared to simply one half of the second utterance. There is finally an issue of how to handle complex sentences where there is a subordinator which is causing a logical relationship. This would require a specialist dataset and again is not simply about Linguistics but similarity, after having identified the relationship.

For the final investigation, the next chapter, instead of pursuing these points, will look at another special case: is looking at sentence similarity and opposites. Despite the size of the MSRP, the selection of the pairs still does not include any opposite pairs and so the dataset is not used again in this thesis.

13.9 Conclusions

Although, SCAWIT gave slightly better classification accuracy than SARUMAN, this is lower than would have been expected from the observed improvement in the core experiment. This appears largely to be due to intrinsic difference between semantic similarity and paraphrases. Because the MSRP is using very similar sentences for the non-paraphrases, much of the advantage in superior similarity measure suggested from the conceptually more correct threshold is not translating to paraphrase identification.

While SCAWIT has the potential to be of use for automation for paraphrase identification as it is giving solid classification accuracy and significantly better results than the baseline, it is equally clear that specialist supervised algorithms are capable of better performance because they consider many different factors rather than the single metric that is output from sentence similarity.

Two corpus based sentence similarity models, IISIS (Islam and Inkpen, 2008) and SSA (Hassan, 2011) did perform better on the dataset than SCAWIT. This is in part due to issues arising with the handling of large compound sentences by the SCAWIT algorithm but with a threshold which would indicate that the improved classification is not necessarily from more accurate similarity but possibly from slightly better resolution for a small number of sentence pairs (~1.2% or 20 pairs). Although this improvement is inside the error bars.

14.0 Handling Opposites

14.1 Introduction

Although, the core experimentation and development had already met the objective to show that a sentence similarity model could be greatly enhanced following a Linguistic approach, there is still one further improvement that will be made to SARUMAN.

The Linguistic approach highlights potentially, a very significant gap in the field of sentence similarity and that is how to deal with comparing ideas that are opposites. This is an area that has already been mentioned at several points in this thesis. Starting out by defining the logical relationships in section 2.7.4 and in chapter 3, a new similarity scale was first presented alongside the opposites dataset that will be used for the experiments in this chapter.

This chapter again adds a new Linguistic component to the sentence similarity model (SCAWIT). It represents the final developmental increment of SARUMAN in this thesis and functions as a final validation of the performance of the model and the power of following a Linguistic approach.

Opposites are an important Linguistic and semantic concept which has not been examined before with sentence similarity. This chapter details the changes needed to include opposites to sentence similarity and evaluates the new implementation of SARUMAN with these concepts upon a specialist domain (opposites dataset).

The new version of SCAWIT, is called SANO and is currently the only sentence similarity model that is specifically able to handle opposites, in a manner that logically reflects their impact upon similarity.

14.2 New Similarity Scale

Oppositeness, like synonyms, is a relationship that directly relates to how similar the meanings of two things are to each other. For two concepts to be opposite, they must share a common scale between them and the opposites represent extreme ends of the scale.

This shared scale means that the connection between a pair of opposites is higher than two completely unrelated sentences. To this point, it would mean, for semantic similarity that opposites would not score the minimum similarity. This poses an interesting question of: should an opposite be represented with a semantic similarity measure?

Semantically, opposite meanings are further apart than unrelated terms. If opposites were scored as the minimum value on the scale of 0.0 to 1.0 then this would cause an issue with completely unconnected sentences, either being indistinguishable from opposites or scoring more highly than would be consistent with earlier human experiments.

However, the difference semantically between a pair of opposites is further apart than two unrelated terms. This is accounted for by using a new scale ranging from -1 to 1. Here the magnitude reflects the connection between the two ideas and the sign whether or not the meanings are opposites, or acting in the same direction. The opposites dataset to be used for the experiment and evaluation in this chapter was rated for semantic similarity using this new scale.

14.3 Types of Opposites

In section 2.6.4, the idea of oppositeness was described as functioning as one of 3 relationships:

- Antonyms - directly opposite words.
- Negatives - Where a clause or sentence is inverted from the inclusion of a negative.

- Inversion - where changing the order of the clauses relative to the verb causes the overall meaning to function as an opposite.

Although, there are fundamental shifts conceptually between SARUMAN and when using opposites, nonetheless, the framework can easily incorporate this Linguistic concept. It is possible to directly include the changes from SCAWIT in order to include opposites with relatively few changes.

14.4 Word Meanings for Opposites

The new scale for the overall output also means that the output of the word similarity module must also be adjusted.

The first step is to include a representation of an antonym relationship between two words within the knowledge database. WordNet contains tags indicating that an opposite exists for a word, (and indeed were included in the OMIOTIS algorithm (Tsatsaronis et al., 2010) when finding shortest path). This would allow for a hypernym chain similar to those of adjectives to be built.

However, instead of adding an extra node to the root of the chain, this time the root node itself will be altered. A unique mapping is used for each type of opposite, using the complement of the letter when ordered from a to z in lower case.

If numbered incrementally from a = 1 to z = 26 the antonym equals the letter equivalent to 27, minus the normal type indicator. An antonym of a noun would be change from 'n' to 'm' as $27 - 14 = 13$ and 'm' is the thirteenth letter of the alphabet. Therefore, a noun meaning linked to an opposite labelled as say "n1234" would become "m1234".

All the opposites included in WordNet were added to the database. One significant addition was made in response to the test domain which was the opposites "hero" and "villain" which were used as an exemplar in the construction of the dataset (section 3.7).

14.5 Word Meaning Similarity Module

If using the standard algorithm, then the meaning compared against its opposite would give a value of 0. As well as making direct comparisons, an additional comparison is made for the antonyms using its complement. This time instead of only comparing "m1234" against "n157" then a second comparison is made for "n1234" against "n157". The score for the meaning similarity is for such a comparison finally negated by multiplying by -1.

One final critical change is needed which is when determining which meaning pair to use, based purely on score. The previous greater than comparison to find the maximum would never use the negative value. So instead the magnitude is used for comparisons before using the highest number. So now -0.75 is selected over 0.6 but not 0.8. Where the two magnitudes are identical but one is positive and the other negative then the positive value is selected.

Assume two nouns: one with 4 meanings ("N1278", "N14649", "M1245" and "M137"); and the other with meaning "N1457". So two of the meanings are antonym relationships. "M1245" is an antonym of words with the meaning "N1245". An antonym compared directly with a normal meaning would give a similarity of 0.

	Antonym	N 1 4 5 7
N 1 2 7 8		0.25
N 1 4 6 4 9		0.419
M 1 2 4 5		0
M 1 4 7		0
M 1 2 4 5	N 1 2 4 5	-0.25
M 1 4 7	N 1 4 7	-0.571

As well as direct comparison, the noun meanings of the antonym group is found. The maximum similarity using PoW occurs with "N147" and "N1457" giving a value of 0.571. However, since there was an antonym involved this is converted to give a negative for "M147" to "N1457". As it has the highest magnitude, the comparison between the two

words is -0.571.

14.6 Opposite Verb Clauses

The opposites of words made a very minor alteration to the model. However, the word interaction means that it is important to determine the function to the overall sentence.

This requires the classification of the function of each clause pair. As well as individual words being opposites, the clauses themselves can be diametrically opposite to one another. It is also possible for clauses to be weakly opposite where only part of the meaning is opposite as opposed to the whole clause.

In the sentence pair:

"I love you"

"I hate you"

The main verb clauses are opposite because the the verbs are direct antonyms.

In addition to antonyms, a negative can make the whole clause invert its meaning:

"I love you"

"I don't love you"

By definition a negation inverts the similarity, in this case the verb clause comparison, is multiplied by -1.0.

There is one exception to this rule and that is where the negative is part of a question.

"Wouldn't you like to go to the theatre with me?"

"Would you like to go to the theatre with me?"

Here, the questions are nuanced but essentially function the same, both could be answered by "yes" to mean the that responder wanted to go to the theatre with the questioner.

Another special case is when there is both antonymy and negation in the same clause pair.

"I hate you"
"I don't love you"

Where both are direct opposites for "I love you", they are not synonyms for each other. While similar they are not the same, but if were simply double inverted the sentences would score a similarity of 1.0.

In this instance a weakening of the similarity is included by treating the negation as an extra property, such as was done with the OPTIC representation (section 9.1) and so the meanings score below unity.

14.7 Other Opposites

The most important clause comparison is the main verb clause but there can be opposites and negation in other clauses. "Not" can function for the whole noun clause but is rare and normally coupled with another clause.

"I enjoy not waking-up at 7 am."
"I enjoy waking-up in the morning"

Again just as with verb clauses the overall clause is adjusted. Often weak opposites will occur:

"I walked quickly to school."
"I walked slowly to school."

In this instance, the combination of the clauses have already been reduced by the inclusion

of the opposite pairs. One of the values would be negative, reducing the magnitude similarity of the verb clause.

However, a negative could have functioned on just the adjunct so as to become:

"I walked not slowly to school".

Now rather than functioning on the whole clause it simply functions on the adverb comparison within the clause in a similar mechanism as if for the whole clause.

14.7.1 Inversion

Inversion is caused through having the subject and object function reversed in a sentence. Not every reversal of function of the sentence leads to a sentence becoming an opposite.

"The blue team beat the red team."

"The red team beat the blue team."

Using the verb beat it is clear that the above pair of sentences cause a contradiction to occur. The opposite is perhaps less obvious but nonetheless exists, as it would be possible to view the result as being on a scale with a draw being a possible result. The reversal of the subject and object does not always lead to an opposite nor a contradiction:

"The blue team drew [with] the red team."

"The red team drew [with] the blue team."

An inversion can lead to a weak opposite as well as pure opposite.

Any stative verbs can have the order of the subject and object clause reversed with no change to the meaning of the sentence:

"The man stands tall."

"The tree is green."

The extra information to make this distinction was already included in the extra level added to the verb hypernym structure when converting WordNet for use with the mathematical model. By examining the second value in a verb hypernym chain a verb clause can be classified whether it is stative or not.

The information for making these judgements was already being added by the parser and it was a simple matter to include the extra component to the model.

The next issue to look at is how the clauses combine together. If simply using the SCAWIT algorithm then the simple sentence with an opposite verb clause and identical subject and object clauses would score 0.0 rather than -1.0.

A decision has to be made for opposites as to whether to combine each clause as a magnitude or a directional value and if the overall similarity needs to be negative.

When the order of the sentences is the same but with a single opposite object clause, it does not cause the whole sentences to function as an opposite.

"James knew a woman"

"James knew a man."

Although a demonstrative clause allows for a single noun clause to function as if it were a sentence itself, it can allow for opposite sentences, due to antonyms for the object clause:

"It is hot."

"It is cold."

The subject clause can form a weak opposite.

"Everybody likes James."

"Nobody likes James."

14.8 SANO Algorithm Module

Despite the change of sign in the similarity, essentially the SANO algorithm is the same as the SCAWIT algorithm with an additional change to judge whether the overall similarity between units should be negative to indicate an opposite relation.

The SARUMAN algorithm uses a square root in its formulas and so cannot cope with negative values. As a result it is necessary to make minor adaptation to formula [6.4] and the combiner also determines the sign of the overall output.

The proximity of a pair of meanings remains unaltered but rather than just having a threshold of 0.2, values below -0.2 also will be considered matched to one another.

If $(\mu_1 + \mu_2) < 0$

$$S_{topic} = -\sqrt{|\mu_1 * \mu_2|}$$

else

$$S_{topic} = \sqrt{|\mu_1 * \mu_2|}$$

[14.1]

Equation [14.1], replacing [6.4], takes the modulus of the weighted means for each sentence and returns a value the same as if there were no opposite values. It is necessary to take the modulus as for weakly opposite sentences the comparison of sentence 1 to sentence 2 could be oppositely signed to one another.

It is also necessary to adjust how the topic and interaction combine with the SARUMAN algorithm when the topic similarity is negative so equation [6.8] becomes:

If $S_{topic} < 0$

$$S_{sentence} = S_{topic} * ratio - S_{interaction} * (1 - ratio)$$

else

$$S_{sentence} = S_{topic} * ratio + S_{interaction} * (1 - ratio)$$

[14.2]

This change is needed because the interaction similarity is only measuring the functional similarity and was directionless, so needs to be aligned to be in the same direction as the topic.

The adapted SARUMAN algorithm is used just as before for comparing the basic clauses and for when there are multiple PSVO structures (preposition clause, subject clause, verb clause, object clause) to be combined.

The SCAWIT combination of the clauses distinguishes between the sign and magnitude of the comparisons. The oppositeness of the PSVO alignment between two sentences is judged as a whole, via the verb clauses comparison.

Antonymy would already be indicated via the value from the clause comparison from the output of equation [14.2] and would have resulted in a negative value for the clause comparison.

The oppositeness from the negation of the verb clause or via inversion are added by a set of logical checks. Where an additional opposite relationship is found then the sign indicating whether the PSVO comparison is opposite is altered (via the sign main verb clauses comparison).

When combining two negatives to make a positive for the sign of the PSVO comparison then the verb clause has its similarity magnitude slightly lowered as if a new distinct property were being added. Otherwise when the judgement that inversion of negation is present, then the sign is changed via multiply by -1.

The individual score for the PSVO comparison is made using the modulus (magnitude) of the verb clauses comparison. This then combines identically to how SCAWIT was doing as detailed in chapter 10. The final output is given the sign of the verb clause.

14.8.1 Identifying Negation and Inversion

The judgement as to whether negation has occurred in the PSVO comparison, is very simple and already included in the information being added by the parser. The verb clauses have already been classed as whether they were negative or not. So it is simply a test to see whether only one of the two verb clauses being compared is negative and not a question. If this is the case then an opposite from negation is added as discussed above in section 14.8.

Other special cases can arise when either the main verb clause comparison is negative or when a high similarity (greater than 0.7 similarity) and non-stative.

Whether or not a verb is stative is already encapsulated in the encoding of the verb meanings from the WordNet ontology described in section 6.3.1. The second digit of the meaning structure already includes whether the verb is stative based upon the value stored in the WordNet database.

Inversion is judged to happen when both the following occur:

- 1) The main verb clause comparison is positive and greater than 0.7 magnitude and neither of the main verbs are purely stative.
- 2) The subject and object clause more strongly cross compared, having greater than 0.7

similarity then inversion happens.

3) Else unless a question inverted from a negative as opposed to an antonym where the main verb clause comparison is negative.

14.9 Results

The modified version of SARUMAN's results on the opposites dataset (section 3.4.4) - labelled SANO (SARUMAN Antonyms Negatives Opposites) can be seen in table 14.1. Figure 14.1 shows the results graphically. The ability for SANO to identify the opposite pairs potentially visually overstates the performance of the model. Pairs 7 and 46 show a weak negative score whereas the human score was positive, together with some other values not very close to the human values. However, it is still the case that a strong performance is being shown with most of the values reflecting that a good overall approximation is achieved. SANO's strong performance is reflected by the Pearson's correlation (PCF) of 0.906.

Run	Sentence 1	Sentence 2	Human	SANO
1	Matthew gave the charity money.	Matthew donated to the charity.	0.89	0.61
2	I love ice-cream.	I do not love ice-cream.	-0.9	-1
3	I cannot paint horses.	I drew the picture.	0.29	0.20
4	I do not run unless I have too.	I run all the time.	-0.07	-0.31
5	I won the race.	I will win the race.	0.71	0.42
6	The favourite should win the race.	A newcomer could win the race if the favourite falls.	0.49	0.27
7	The favourite should win the race.	Should the favourite fall a newcomer could win the race.	0.49	0.03
8	You can do it.	You did it.	0.72	0.84
9	The big man would make a good king.	For the sake of the country, Mark must rule.	0.31	0.54
10	I must finish.	I might finish.	0.55	0.72
11	Cats enjoy sleeping.	A cat enjoys sleeping.	0.98	0.97
12	The oak tree will grow in the spring.	An asteroid will hit soon.	0.08	0.52
13	Do the dishes.	Wash-up.	0.92	1
14	Hear the birds singing.	Smell the scent of the roses.	0.19	0.05

Table 14.1a: Opposites Dataset and SANO (cont...)

Run	Sentence 1	Sentence 2	Human	SANO
15	The dog fighting in the street is black.	The dog fights in the street.	0.56	0.50
16	The man is too old.	The woman declared, "the man is too old."	0.59	0.36
17	Elephant	Mouse	0.52	0.71
18	The big grey animal	An elephant	0.85	0.80
19	The black cat is drinking from a saucer of milk.	The white cat is drinking milk from a saucer.	0.74	0.46
20	The small elephant.	The Elephant.	0.82	0.96
21	The old teacher gave the boy the money,	The professor gave treasure to the girl.	0.65	0.76
22	You, the Roman people, are an inspiration.	The Romans are an inspiration.	0.78	0.78
23	The woman ran.	She goes quickly.	0.71	0.36
24	The warrior whipped the slave.	The warrior used the whip on the slave.	0.86	0.77
25	The slave was whipped by his master.	The slave was not whipped.	-0.84	-0.75
26	The hero won the fight.	The villain won the fight.	-0.85	-1
27	The hero won the fight.	The hero lost the fight.	-0.92	-1
28	The hero slew the villain.	The villain slew the hero.	-0.87	-1
29	Cats do not like to be stroked backwards.	The man yelled with pain when he hit his thumb.	0.13	0.40
30	The cameraman shot the wedding.	The sniper shot his enemy.	0.29	0.62
31	The dog ate the bone.	The bone was eaten by the dog.	0.98	0.95
32	It was raining.	I hate daylight.	0.24	0.21

Table 14.1b: (...cont) Opposites Dataset and SANO (cont...)

Run	Sentence 1	Sentence 2	Human	SANO
33	The cat drank lemonade.	My good friend only eats bananas.	0.15	0.44
34	The ice-cream children like the best is vanilla.	The children wanted ice-cream.	0.58	0.21
35	What is the answer?	The answer is 6.	0.46	0.71
36	Won't you come to the game?	Will you come to the game?	1	1
37	Don't you dare do it!	Don't you dare do it?	0.79	0.94
38	The butter had been left out too long.	The butter has been left out too long.	0.99	0.95
39	The first time that I saw her I knew I was going to marry her.	I am engaged to my fiancée.	0.64	0.23
40	The water makes them thirstier.	The dry desert is very hot.	0.29	0.29
41	He imagined winning the race.	He wins the race.	0.69	0.76
42	He should win the race.	He imagined winning the race.	0.59	0.81
43	The slow car went to London.	The man drove slowly to London.	0.84	0.75
44	The man shot the rabbit.	The hare was injured by the boy.	0.67	0.71
45	He whipped the slaves.	He used the whip on the horse.	0.78	0.77
46	My donation was welcomed by James.	I gave James the money.	0.71	-0.17

Table 14.1c: (...cont) Opposites Dataset and SANO

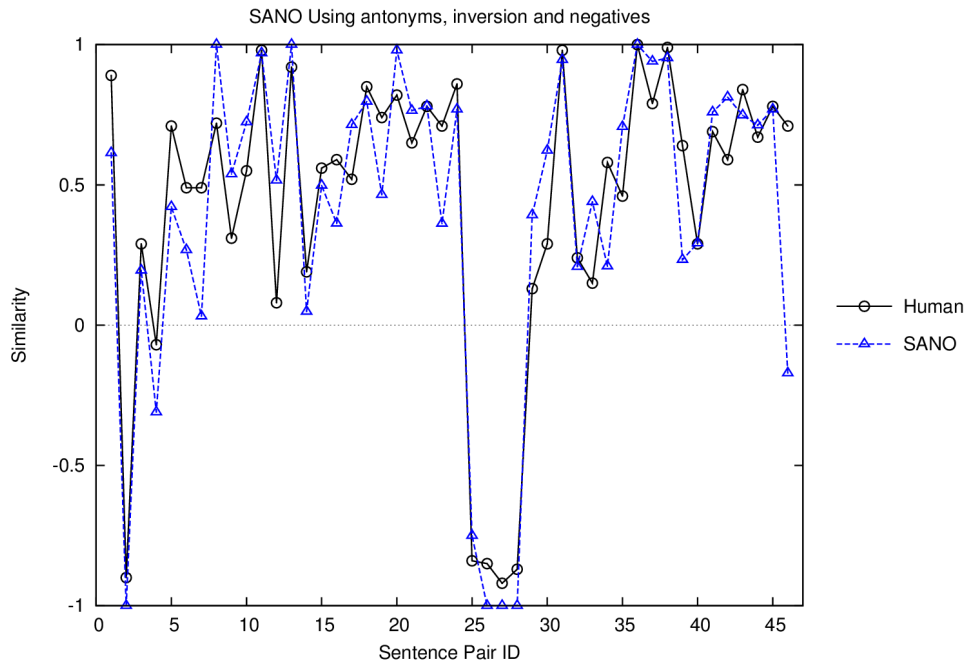


Figure 14.1: SANO for the opposites data set

The dataset is fundamentally a different scale and when processed with either LSA or SCAWIT then only a weak negative correlation is resultant. With LSA only managing -0.116 PCF and SCAWIT fractionally better at -0.056 PCF. The weak negative correlation is a sign of a very poorly performing relationship. Where a strong negative correlation would still represent a strong connection between the data being correlated, a weak correlation is only a little better than what could occur with random data.

Although, LSA was identified as being unable to cope with negations, it was regarded as a complete algorithm and originally using a scale of -1 to 1 (Deerwester et al., 1997). As such, it is marked that the idea of opposites has been overlooked by the approach. The examination of the basic algorithm, is used to highlight that the new scale can lead to a very differing performance from standard similarity approaches.

The number of negative pairs, although small, strongly dominate the correlation so a large difference in performance between SANO and the models not including these sentences

conceptually, is unsurprising. Although, conceptually beyond LSA as it has no knowledge of the clauses and completely omits negatives such as "not", it would be possible to include human tagging with the input to indicate where a pair of sentence's were opposites. This could then allow the score from a model to be negated to approximate the scale. This is done using the human negative scores as tagged sentence pairs for both SCAWIT and LSA.

Model	Pearson's	Spearman's
LSA + human tagged	0.837	0.599
SCAWIT + human tagged	0.928	0.814
SANO	0.906	0.802

Table 14.2: Summary information for SANO and opposite tagged LSA and SCAWIT.

Table 14.2 gives the summary information for both LSA and SCAWIT. There is a dramatic improvement from tagging for the LSA correlation which jumps to a Pearson correlation of 0.83 and a Spearman's correlation of 0.60. This still remains significantly lower than the fully automated SANO results (PCF = 0.91 and SRC = 0.80). This indicates that the underlying SCAWIT algorithms are still performing more strongly than the LSA algorithm. Indeed, when repeating the tagging with SCAWIT then a PCF of 0.928 occurs.

SANO still had some issues with identifying weak negatives which, in part, is due to not handling the conditional "unless" properly. Sentence pair 46 led to confusion with failure to handle the combinations of the meanings properly.

The tagged version of SCAWIT shows both an ideal limit of the model for the dataset and indicates that the dataset is perhaps easier to process than the thirty pairs dataset.

14.10 Conclusions

This final experiment both showed the power of the framework and of the Linguistic approach. With comparatively minor changes to the SCAWIT model, it was possible to construct a sentence similarity model that was capable of handling opposites and a new

sentence similarity scale.

SANO is the only sentence similarity model that is capable of handling opposites and so it was not possible to directly compare it to another model. However, even when using human tagging it can be seen that SANO was still giving notably better correlation for the opposites dataset.

The opposites dataset is not designed to be representative of the frequency of the occurrence of opposites expected from sentence pairs in general (section 3.5). Therefore, the performance in terms of the correlations achieved for the dataset do not necessarily reflect the overall performance on any input. What is clearly shown is that by adapting SARUMAN to include Linguistic concepts, including opposites, that far higher accuracy on certain types of sentences can be achieved.

Negatives are amongst the most common words in the English language, even in written corpora. So while the difference in magnitude of the accuracy is a clear exaggeration, this is still a very important stage in considering sentence similarity. It is very clear from a Linguistic perspective that the antonyms, negatives and inversion are critical to the meaning of the sentences and this has shown that the ideas can be incorporated into a sentence similarity model, through a simple set of conditional rules.

The new sentence similarity scale was easily handled by SARUMAN with only minor changes to the model and it managed a very striking difference between a normal similarity measure on the dataset, with an increase in Pearson's correlation of more than 1.0. Although it was shown that tagging would allow LSA to perform much closer to the SANO model, it is not possible to easily automate this knowledge for LSA without considering several Linguistic concepts completely absent from LSA.

This still small dataset also shows that the underlying SCAWIT method in SANO is showing repeatability. Again giving statistically significant better correlation than even the tagged version of LSA. The introduction of a new similarity scale might be more significant than the fact that a strongly performing automatic sentence similarity model was produced. Just as was the case with paraphrases, there are likely to be different specialist applications where one of the similarity scores is better suited than the other.

The results strongly re-iterates the findings from the core experiments, that the Linguistic concepts can improve sentence similarity.

While there are still many other Linguistic rules, such as conjunctions and subordinators which are not fully handled by SANO, the steps which have been added are a substantial development to sentence similarity and show the potential to produce a more accurate and speedy sentence similarity model.

15.0 Conclusions

15.1 Introduction

This research set out with the objective to show that it was possible to improve a sentence similarity model from the inclusion of Linguistic concepts and that objective has been met from the results obtained in this thesis.

Linguistics is focused on describing how the patterns in English combine to give meanings for human understanding. The first step toward accomplishing the objective of the research was to adapt fundamental Linguistic concepts for comparing the meanings of pairs of sentences.

In addition to having an underlying knowledge source from which the similarity between words in a fixed vocabulary could be found, three other key areas were identified as important to sentence similarity. These were described as : topic, word interaction and context. These concepts were used to build the modular Linguistic framework that is the foundation of the experiments and the creation of the sentence similarity models.

The ability to gradually introduce fundamental Linguistic elements to a sentence similarity model adhering to the framework, allowed the core experimentation to isolate the impact of individual changes to a sentence similarity model. The isolation of the effects of the particular concept being added, leads to minimal noise in the experimentation allowing for the use of a smaller dataset.

Having outlined the experimental method, the next stage was the construction of a mathematical model to which the components could be added to in the future. The new model, SARUMAN was created although re-using many features found in other sentence similarity models. SARUMAN used WordNet as its knowledge base WordNet because ofwith its ontological relationships. The outline of how to use all of the parts of speech within WordNet was given.

SARUMAN's algorithm allows for vectors of values giving the similarity of each word but and can use the combination of vectors of different lengths. This makes it distinct from the other knowledge based similarity models and removes the need for additional artificial steps, which would move it further away from the Linguistic approach desired.

SARUMAN was shown to give comparable results to other mathematical models on the standard dataset used in the Literature where it was showing strong correlation. In contrast, the model could be seen to struggle with the new ten pairs dataset. This was also true of the other sentence similarity models using WordNet as their knowledge base.

Having established a benchmark with SARUMAN, the next stage of the investigation was focussed on disambiguation. The framework did not have a direct correspondence to context but rather had introduced context via weights.

It could be seen that when using a person to determine the intended meaning for each word then adding this information to the sentence, that disambiguation of meaning showed a substantial improvement in performance over SARUMAN. This showed that having a single intended meaning for each word in a sentence, could greatly enhance the comparison, compared to taking the highest similarity as is done with other sentence similarity models when the sentence ambiguity must be resolved.

When using the part of speech to exclude the inappropriate meanings which can more reliably be found by automation than the meanings, it was found that the sentence similarity again could be improved, although to a lesser extent than for the meanings,

An automated approach at disambiguating the meanings of words in their context, using the associations formed from the WordNet definition for each word showed some success. When the meanings were restricted to the type first, an improvement in the similarity resulted, but in some instances this was done without giving the same meanings as had been obtained from the human tagging.

The automation of word sense disambiguation was struggling in several respects but this is perhaps one of the most challenging areas for automation. Partial success was achieved at finding the right meaning using the definitions from WordNet, but limits in the information

in the definition and inconsistency in the meanings structure obtained from WordNet (meaning that it was notn't currently aiding the algorithm).

While meaning disambiguation showed promise, the reward to the algorithm from selecting the right meaning, as opposed to a close meaning, was small and even in some occasions because of the ontological structure in WordNet being incomplete - worsening the performance not aiding it.

Therefore, when moving forward only the type disambiguation was used but nonetheless it was still clear that disambiguation is an important method to improve sentence similarity by removing unwanted comparisons. The issue with meaning disambiguation is largely the issue of struggling to automate a task that can be dependent on understanding the underlying meaning.

Having shown that the context and disambiguation improved SARUMAN, the next area which was examined was that of word interaction. The words in a sentence do not operate in isolation but combine together to form more complex meanings. The most fundamental basic unit above the word level is the clause. SARUMAN had been using a free comparison where any word was freely compared to any other word in the complementary sentence.

Some of the provisional investigation using properties to represent meaning suggests that this issue could still potentially be improved, but even should this not prove to be the case, the framework allows for humans to tag the intended meanings. While for a single sentence pair this is a prohibitive overhead were the same sentence be compared many times, then it could have the meanings and part of speech tagged and stored for the future use.

The next stage of development introduced combination of meanings into single units and advanced word interaction. This involved making a conceptual change to the meaning representation from the hypernym chains to a properties model. This version of SARUMAN introduced merging clauses and was given its own name of SCAWIT.

A new version of the word meaning similarity module was enacted using the properties of words formula from Pearce et al. (2011). While much of this potential was unrealised as

the bulk of the vocabulary was the still using the same structure as WordNet, a provisional investigation using by hand creation of a set of properties showed that this could potentially be a significant improvement to SARUMAN.

SCAWIT had been shown to be a significant improvement over a mathematical model using WordNet's lexical database as its knowledge via SARUMAN. Even without resolving the potential issue in the knowledge base or with meaning disambiguation, SCAWIT was still shown to be a strongly performing sentence similarity model. While it was not achieving a level of accuracy expected by human judgement, the ability to process 100s of sentence pairs a second means that it could be a powerful aid to partial automation.

SCAWIT consistently outperformed the corpus based method LSA. It showed stronger correlation, on the ten pairs dataset with the noise from differing WMS algorithms, on the 30 pairs dataset and even on the opposites dataset. SCAWIT also outperformed LSA on the MSRP paraphrase identification.

The performance on the specialist domain of paraphrases, showed that while the sentence semantic model was performing well above average that it was not better at the task than some of the corpus based methods, but clearest still was that specialist algorithms were better suited for the task than sentence similarity. These results, and later with SANO on opposites, suggest that some situations where a sentence similarity is less well suited than a specialist application.

The Linguistic approach made it obvious that opposites was an area not currently being adequately handled by the standard sentence similarity metric. SANO's performance on the opposites dataset showed a clear improvement over existing models and substantial conceptual gap in pre-existing models. Not only do the other models not handle opposites specifically, they completely exclude the information that is contributing to oppositeness. While it was a simple extension for SCAWIT to include the opposites, this was because it was already adding much of the relevant structural information from the parser module.

Even were the oppositeness decided by a separate program with human level of judgement, this would have been the same case as when tagging the meanings in the final experiment: the strength of the SARUMAN algorithm still showed improvement.

The result of all these findings is that a Linguistic approach has shown that the field had the potential to improve sentence similarity. The most advanced models produced as part of this research, while still open to further enhancement because of the framework, are showing improved performance on many sentence pairs because of its handling of Linguistics. While the exact expected performance on a domain is not indicated by the experiments, as this would depend on the specific domain being used and the frequency of the features, it is the case that there will be many cases where SCAWIT will outperform the pre-existing models because of this Linguistic approach.

15.2 Contributions

Several key areas of novelty were added from this research. The most significant was in adopting a Linguistic approach and showing how this could lead to a more accurate sentence similarity model.

Adapting some of the core concepts of English identified by Linguistics for the task of sentence similarity enabled the creation of novel modular Linguistic framework. The inclusion of modules that directly related to Linguistic concepts was a very powerful contribution, that allowed for the easy extension and gradual improvement of a sentence similarity module. While this provided a convenient mechanism for evaluating the contribution of an individual concept, it was also shown to aid in the easy evolution of sentence similarity.

A new mathematical model using the framework was created. Although, this reused many proven components found in other sentence similarity models, such as Resnik (1995) importance weights and Li et al. (2003) word similarity algorithm for WordNet relationships, it had distinct differences.

SARUMAN's algorithms avoided the need for any artificial step to be included in order to combine the similarities between the word meanings. It could reduce the similarity scores for each word into a vector which would be the same length as the sentence. This means

that it could directly handle the words in the order that they appeared in the sentence. This removed some of the overestimation occurring within closely related models such as STASIS,

SARUMAN performs solidly with other models but with the advantage of being easily extensible by the virtue of adhering to the framework. The extensibility was clearly shown as it was developed to include disambiguation, word interaction and finally opposites.

SCAWIT represented not only a demonstration of how more complex word interactions could be used for merging meaning to improve the performance of SARUMAN, but showed itself capable of outperforming many pre-existing sentence similarity models.

SANO became the only sentence similarity model able to explicitly handle opposites (negatives, antonyms and inversion). While models such as OMIOTIS had included the antonym relationship pointer in their word similarity, they had not done so in a manner that recognised how opposites behaved logically for meaning. Negatives are amongst the most common words in English and can have a dramatic effect on the semantic meaning, yet existing models were excluding these words as stop words and entirely excluding them.

It would also not be easy to reintroduce the concept without these models also being adapted to include significant Linguistic information. Yet, because of the Linguistic approach and the framework the changes needed to SCAWIT were minor.

The inclusion of word interaction to a sentence similarity model was another key addition to sentence similarity. While structures based upon the word order similarity can in many instances handle word interaction, there are occasions where this will fail or even hinder the processing of a sentence because of the more complicated manner in which meaning can be found.

The use of a properties of word representation meaning allowed for an enhancement of the vocabulary handled by a sentence similarity model. Several sentence similarity models had already included the verbs and adjunct ontological structures from WordNet in their algorithm. However, no detail is given as to how they resolved the lack of the root node for the verbs, adjectives or adverbs, when dealing with cases without a pointer to a noun

synonym group. The STASIS dataset contained negligible variation in the verbs and adjuncts meaning that any contribution from this would not show through in the reported results.

This research included an expansion of the WordNet ontologies to include the other parts of speech. Auxiliary verbs were encoded solely through considering the meanings as properties. Individually, these changes were not tested to see whether a significant improvement would result but they led to the natural evolution of opposites.

While the framework and development of the SCAWIT algorithm represent substantial potential improvement to the field of sentence similarity, it is perhaps the creation of a new sentence similarity scale that will prove to be most influential.

The creation of the scale raises questions about what direction sentence similarity should head as to whether the opposite scale should be considered as a special handling of similarity or whether it should replace the standard similarity.

Essentially SANO retains the SCAWIT algorithm at its core and if there are no opposite relationships within the sentence pair then it should give the same output.

Other key aspects of novelty showed potential but have only been evaluated in a provisional manner. These included the properties representation of meanings. The by hand experiments showed that these might work more strongly with SCAWIT than the cruder meanings, but there would still remain disambiguation issues to resolve before this could be regarded as conclusive.

The automatic disambiguation showed some success but it is not clear that partial success in disambiguation from using WordNet definitions was always going to be beneficial and was not included in the current version of SCAWIT. With the change to the properties model, it is possible that both the accuracy and the speed of the disambiguation could be improved over the WordNet definitions.

The other component which has not been formally been benchmarked was the sequential parser. It was seen that the approach enabled the patterns from Linguistics to be encoded in

rules and that a fast parser was obtained. However, because it is using slightly different outputs from the standard parsers and its focus for handling disambiguation is focused on the task of sentence similarity, it would require adaptation for testing.

The accuracy of a parser, however, is largely determined by its rules and the speed of the parser should be possible to implement with any rules based parser and is likely to be faster.

15.3 Key Findings

It was found that Linguistics could not only be adapted for use with a sentence similarity model but that the inclusion of fundamental Linguistic concepts led to significant improvement to sentence similarity.

Unlike other sentence similarity models the use of a modular extensible Linguistic framework does not lead to a closed sentence similarity model that can be considered complete. It was shown that individual Linguistic concepts could be gradually introduced to SARUMAN with only minor changes because of its adherence to the framework. Even, SANO, the final version in the development of SARUMAN, has many other features that could potentially be added to it based upon Linguistic observations because of the configuration of the framework.

By virtue of the Linguistic approach, perhaps as many new avenues of research have been opened as questions answered by the research. However, many of these represent rules that would only apply to specialist classes of sentences rather than the very common features that might be applicable to any sentence, that were added as part of the development of SARUMAN,

An entirely new sentence similarity scale was created to deal with the idea of opposites and sentence similarity. When considering the issue of opposites it becomes indisputable that the inclusion of Linguistics has allowed the model to go beyond the capability of other

sentence similarity models. This is the first time that opposites have been considered with sentence similarity and so represents a new area for sentence similarity to investigate further.

The pursuit of introducing Linguistics to the sentence similarity resulted in several areas of novelty being included in the model. This included: potential automatic disambiguation; a sophisticated new sequential parser and an alternative structure for representing meanings of words for computers. The core inclusions though from a research perspective were disambiguation by part of speech, clauses and the ability to merge clauses. In addition the conceptual introduction of opposites into the domain of sentence similarity.

It was successfully demonstrated that it was possible to incorporate common Linguistic concepts into a sentence similarity model and see improved accuracy from each inclusion. The results showed that for some types of sentences that the Linguistic model showed significant improvement over the pre-existing methods.

When further tested in domain of paraphrases again strong correlations were obtained. However, it was also found to be the case that the idea that a specialised model can be significantly better even on semantic similarity related tasks such as the paraphrase identification.

Although the experiments show that the model was capable of producing strong correlations at a rate of several 100 pairs a second, it was also seen that a general purpose sentence similarity model including Linguistics did not automatically display better performance on all tasks.

It would be reasonable to state that the model including Linguistic concepts is superior to the other existing similarity models giving a high probability that its output is nearer to the human rating. Although it is not guaranteed that the handling of certain types of sentences better will give a significant improvement of performance on every dataset, it would be reasonable to view the sentence similarity model as superior for the general case.

Even with these substantial introductions, this still does not mean that the sentence similarity model should be considered as complete. It uses rules which represent the ideas

of Linguistics but not necessarily the only or optimal method.

Because there are no defined methods for converting Linguistics into simple rules, that can be interpreted by a computer, this means that the implementation within the model are suboptimal having inherent approximations.

Additionally, there are many further Linguistics concepts and observations which have the potential to be added to SARUMAN. It can be seen from where direct human knowledge was used for disambiguation, that Linguistically there is scope for significantly better performance but that automation and development issues prevent this from being realised with a computational model.

Nonetheless, the key purpose of the research was to demonstrate that Linguistics could be used to produce a better sentence similarity model than a purely mathematical one and it has been shown that this is the case, with the statistically significant improvements from the core experimentation.

The final models which have been produced, whilst not conceptually closed, still have usefulness as applications. The ability to process a large number of sentences far faster than a human, whilst giving a strong correlation with the human rating even though only an approximation, offers a powerful tool for automation.

With it being shown that real situations can be better suited to specialist applications rather than the general case, when considered as part of an application much of the future development is best suited to being adapted in its operational environment. This is particularly relevant when considering the issues of vocabulary where many disambiguation problems which the computer has to face, involve obscure definitions of polysemous words which may not be present in all domains.

15.4 Future Work

Several areas of interest have been highlighted as a result of the research presented in this

thesis. With sentence similarity being a field which is both still in its infancy and with significant potential for automation, it is not surprising that a large number of areas can be identified for future research.

The two core approaches for further advancement can be thought of as a purely academic approach and a technological application. The latter would be the preferred option as it would allow for both the continual improvement of a model and allow some of the benefits of sentence similarity to be realised.

The framework means that any model that adheres to the framework is extensible and its modules can be evolved individually. This trait means that the model could continue to have further Linguistic concepts added to it to cover further situations. However, there are several areas where large resources would be required in order to investigate the Linguistic areas not already covered in this thesis.

The biggest unanswered question is how best to deal with larger sources of information such as at the document level. This is not just important to sentence similarity, especially with respect to dealing with conjunctions, but to several other areas too.

Sentences represent a convenient division to use for documents but it is also possible to divide longer compound sentences into simpler sentences. In several applications it may prove to be the case that the document is best categorised using phrases rather than sentences. This then would mean that the model using the framework would need to have a stage added prior to its execution.

15.4.1 Tagged Corpora

There is a dearth of sources that contain human rated knowledge for sentences. While the ultimate aim is to automate these tasks, there are situations where there exists perfect ambiguity for a machine. This means that an estimate of probability of each of the possibilities is needed. This requires having a source of data which is consistently tagged with respect to the disambiguation of meaning and clauses. Due to the scale of such data, it

is likely that this too would require a level of semi-automation.

15.4.2 Database of Properties of Words

The final models in this thesis are using a word meaning similarity algorithm based upon the properties of ideas. A word is normally regarded as a single word divided by spaces, however, the existence of collocations means that a crisp division cannot be made between words. It would be useful to have a database for the properties of words on a similar scale as WordNet. Automation suffers with the issues of disambiguation of sources but with tagged corpora may be possible from some of the words.

15.4.3 Greater Collaborative Resources

It would be useful to the task of benchmarking sentence similarity models if there were a large comprehensive dataset. There is the desire to definitively judge the accuracy of sentence similarity but it would take hundreds of thousands of tailored sentences with expert consensus as to the similarity score to produce such a benchmark. While the advantage of such a store of information to sentence similarity is obvious, it is equally unlikely that the resources needed to produce such a dataset will be forthcoming without significant technological breakthroughs increasing the interest in the field of sentence similarity.

It might be possible as sentence similarity begins to be used for more real applications, to build common online resources with provisional scores to be gradually built upon with the eventual aim of finally producing a suitable benchmark dataset. This approach, however, has been complicated by the existence of other scales such as presented in chapter 14 for opposites or when dealing with tasks linked to relatedness.

References

P. Achananuparp, X. Hu and X. Shen: "The evaluation of sentence similarity measures," *In Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery*, Turin, Italy, pp. 304-315, 2008.

S. Banerjee and T. Pedersen: "An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet," *3rd International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 136-145, 2002.

R. Baeza-Yates and B. Ribeiro-Neto: *Modern information Retrieval*, Addison-Wesley, 1999.

J. R. Bellegarda: "Spoken Language Understanding for Natural Interaction: The Siri Experience," *In Natural Interaction with Robots, Knowbots and Smartphones*, pp. 3-14, Springer New York, 2014.

W. Blacoe and M. Lapata: "A comparison of vector-based representations for semantic composition," *Proceedings of EMNLP*, Jeju Island, Korea, pp. 546-556, 2012.

T. Brants: "TnT -- A Statistical Part-of-Speech Tagger," *6th Applied Natural Language Processing Conference*, Sydney, Australia, pp. 224-231, 2000.

E. Brill: "A simple rule-based part of speech tagger", *In Proceedings of the third conference on Applied natural language processing (ANLC '92)*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 152-155, 1992.

C. Burgess, K. Livesay, and K. Lund: "Explorations in context space: words, sentences," *Discourse Processes*, 25 (2-3), pp. 211-257, 1998.

D. Chandran, K. Crockett, D. McLean, and Z. Bandar: "FAST: A fuzzy semantic sentence similarity measure," *IEEE International Conference on Fuzzy Systems (FUZZ)*, India, pp. 1-8, 2013.

G. Chrupala, G. Dinu, and J. van Genabith: "Learning Morphology with Morfetea," *Proceedings of LREC 2008*, Marrakesh, Morocco, pp. 2362-2367, 2008.

K. W. Church: "A stochastic parts program and noun phrase parser for unrestricted text," *ANLC '88: Proceedings of the second conference on Applied natural language processing*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 136-143, 1988.

D. Crystal: *The Cambridge Encyclopedia of Language*, Cambridge University Press, Cambridge, 1987.

D. Das and N. A. Smith: "Paraphrase identification as probabilistic quasi-synchronous recognition," *Proceedings 4th International Joint Conference on Natural Language Processing of the AFNLP*, Singapore, pp. 468–476, 2009.

M. De Boni and S. Manandbar: "The use of sentence similarity as a semantic relevance metric for question answering," *Proceedings of AAAI Symposium on New Directions of Question Answering*, pp. 138–144 , 2003.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landnauer, and R. Harshman: "Indexing by latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259-284, 1990.

P. Denis and b. Sagot: "Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort," *PACLIC 2009*, Hong Kong, pp. 2744-2751, 2009.

S.C. Dik, *The Theory of Functional Grammar. Part 1: The structure of the Clause*, Berlin: Mouton de Gruyter, 1997.

B. Dolan, C. Quirk, and C. Brockett: "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," *Proceedings of the 20th international conference on Computational Linguistics*, pp. 350-es, 2004.

B. Dolan, C. Quirk, and C. Brockett: *Microsoft Paraphrase Corpus*, Documentation, <http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/>, 2005.

K. Erk and S. Pado: "Shalmaneser - a flexible toolbox for semantic role assignment," *Proceedings of LREC 2006*, Genoa, Italy, pp. 527-532, 2006.

S. Federici, S. Montemagni and V. Pirrelli: "Shallow Parsing and Text Chunking: a View on Underspecification in Syntax," In *Proceedings of the Workshop On Robust Parsing. ESSLLI*, Prague, Czech Republic, pp. 35-44, 1996.

C. Feldbaum (ed.): *WordNet an Electronic Lexical Database*, MIT press, 1998.

S. Fernando, and M. Stevenson: "A semantic similarity approach to paraphrase detection," *Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloquium*, UK, pp. 45-52, 2008.

A. Finch, Y. S. Hwang, and E. Sumita: "Using machine translation evaluation techniques to determine sentence-level semantic equivalence," *Proceedings of the 3rd International Workshop on Paraphrasing*, Jeju Island, Korea, pp. 17-24, 2005.

W. N. Francis and H. Kucera: *Brown Corpus Manual*, Brown University Press, 1979.

R. Garside and N. Smith: "A hybrid grammatical tagger: CLAWS4," in Garside, R., Leech, G., and McEnery, A. (eds.) *Corpus Annotation: Linguistic Information from Computer Text Corpora*, Longman, London, UK, pp. 102-121, 1997.

E. Gabrilovich and S. Markovitch: "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," *20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, pp. 1606-1611, 2007.

R. J. Gaizauskas, M. A. Greenwood, M. Happle, I. Roberts, H. Saggion, and M. Sargaison: "The University of Sheffield's TREC 2003 Q&A Experiments," In *Proceedings of the Twelfth Text Retrieval Conference (TREC)*, Gaithersburg, MD, USA, pp. 782-790, 2003.

R. L. Goldstone and J. Son: *Cambridge Handbook of Thinking and Reasoning*, Cambridge University Press, Cambridge, 2005.

"Google search engine," <http://www.google.co.uk>, July 2011.

"Google Ngram data," storage.googleapis.com/books/ngrams/books/datasetsv2.html, 2013

"Guardian website", <http://www.theguardian.com/environment/>, 2013.

S. Hassan: *Measuring Semantic Relatedness using Salient Encyclopedic Concepts*, Ph.D. thesis, University of North Texas, Denton, TX, USA, 2011.

K. Hengeveld and L. Mackenzie: *Functional Discourse Grammar: A Typologically-Based Theory of Language Structure*, Oxford University Press, Oxford, 2008.

A. Hliaoutakis, G. Varelas, E. Voutsakis, E.G.M. Petrakis and E. Milios: "Information retrieval by semantic similarity," *International Journal on Semantic Web and Information Systems*, vol. 2(3), pp. 55-73, 2006

C. Ho, M. A. A. Murad, S. C. Doraisamy and R. A. Kadir: "Measuring Sentence Similarity from Both the Perspectives of Commonalities and Differences," *Proceedings of 22nd International Conference on Tools with Artificial Intelligence*, Lens, France, pp. 318–322, 2010.

R. D. Huddleston, G. K. Pullum, and L. Bauer: *The Cambridge Grammar of the English Language*, Cambridge University Press, Cambridge, 2002.

A. Islam and D. Inkpen: "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Transactions Knowledge Discovery Data*, vol. 2, pp. 1-25, 2008.

J. Jiang and D. Conrath: "Semantic similarity based on corpus statistics and lexical taxonomy," *Proceedings of Research in Computational Linguistics*, Taiwan, pp. 19-33, 1998.

M. P. Kaschak and A M. Glenberg: "Constructing meaning: the role of affordances and grammatical constructions in sentence comprehension," *Journal of Memory and Language* 43, pp. 508 –529, 2000.

S. Kempken, W. Luther, and T. Pilz, "Comparison of distance measures for historical spelling variants," *In Artificial Intelligence in Theory and Practice*, pp. 295-304, Springer US, 2006.

A. Kennedy and S. Szpakowicz: "Evaluating Roget's thesauri," in *Proceedings of ACL-08: HLT*, Columbus, Ohio, USA, pp. 416-424, 2008.

D. E. Knuth: "Semantics of context-free languages," *Mathematical Systems Theory*, 2(2), pp. 127-145, 1968.

Z. Kozareva and A. Montoyo: "Paraphrase identification on the basis of supervised machine learning techniques," *Advances in Natural Language Processing: 5th International Conference on NLP (FinTAL 2006)*, Turku, Finland, pp. 524-533, 2006.

L. Larkey and A. B. Markman: "Processes of similarity judgement," *Cognitive Science*, vol. 29(6), pp. 1061-1076, 2005.

T. K. Landnauer and S. T. Dumais, "A solution to Plato's problem: the Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104(2) , pp. 211-240, 1997.

C. Leacock and M. Chodorow: "Combining local context and WordNet similarity for word sense identification," *WordNet: An Electronic Lexical Database*, pp. 265-83, 1998.

M. Lesk: "Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone," *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC)*, Toronto, Canada, pp. 24-26, 1986

Y. Li, Z. Bandar and D. McLean: "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol.15(4), pp. 871-882, 2003.

Y. Li, Z. Bandar, D. McLean, J. O'Shea and K. Crockett: "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol.18(8), pp. 1138-1150, 2006.

S. Liberman and S. Markovitch: "*Wikipedia-based Compact Hierarchical Semantics with Application to Semantic Relatedness*", Technical report CIS-2010-06, Technion Computer Science Department, 2010.

D. Lin: "An Information-theoretic Definition of Similarity," *Proceedings of the Fifteenth International Conference on Machine Learning*, Madison, Wi, USA , pp. 396-304, 1998.

M. Lintean, V. Rus and A. Graesser: "Paraphrase identification using weighted dependencies and word semantics," *Informatica*, 34(1), pp. 19-28, 2010.

X. Liu, Y. Zhou, and R. Zheng: "Sentence similarity based on dynamic time warping," *International Conference on Semantic Computing*, Irvine, CA, USA, pp. 250-256, 2007.

LSA implementation: "LSA: on-line latent semantic analysis implementation," lsa.colorado.edu, Feb. 2013.

N. Madnani, J. Tetreault, and M. Chodorow: "Re-examining machine translation metrics for paraphrase identification," *Proceedings of 2012 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2012)*, Montreal, Canada, pp. 182-190, 2012.

C. D. Manning: "Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?" *Proceedings of 12th International Conference in Computational Linguistics and Intelligent Text Processing*, Tokyo, Japan, pp. 171-189, 2011.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini: "Building a large annotated corpus of English: The Penn Treebank." *Computational linguistics*, vol. 19(2), pp. 313-330, 1993.

A. Markman and D. Gentner: "Structural alignment during similarity comparisons," *Cognitive Psychology*, 25, pp. 431-467, 1993.

H. Maynard, K. McTait, D. Mostefa, L. Devillers, S. Rosset, P. ParouBek, C. Bousquet, K. Choukri, J. A. J. Goulian, F. Bechet, O. Bontron, L. Charnay, L. Romary, M. Vergnes, and N. Vigouroux: "Constitution d'un corpus de dialogue oral pour l'evaluation automatique de la comprehension hors et en context du dialogue," *Journées d'Etude sur la Parole (JEP)*, Morocco, pp. 357-360, 2004.

T. McArthur (ed.): *The Oxford Companion to the English Language*, Oxford University Press, Oxford, 1992.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajic: "Non-projective Dependency Parsing using Spanning Tree Algorithms," *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, Canada, pp. 523- 530, 2005.

R. Mihalcea, C. Corley, and C. Strapparavo, "Corpus-based and knowledge-based measures of text semantic similarity," *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA, USA, pp. 775-780, 2006.

G. A. Miller and W. G. Charles: "Contextual correlates of semantic similarity," *Language and Cognitive Processes*, vol. 6(1), pp. 1-28, 1991.

J. Mitchell and M. Lapata, "Composition in Distributional Models of Semantics," *Cognitive Science*, vol. 34(8), pp. 1388-1429, 2010.

J. Morato, M. Marzal, J. Llorens, and J. Moireeiro: "WordNet applications," *Proceedings of the Second Global WordNet Conference*, Czech Republic, pp. 270-278, 2004.

D. Q. Nguyen, S. B. Pham, and D. D. Pham: "Ripple down rules for part-of-speech tagging," *Proceedings of 12th International Conference in Computational Linguistics and Intelligent Text Processing*, Tokyo, Japan, pp. 190–201, 2011.

J. Nivre: "An efficient algorithm for projective dependency parsing," *Proceedings of the 8th International Workshop on Parsing Technologies*, Nancy, France, pp. 149-160, 2003.

"NLTK: Natural Language Toolkit website", <http://www.nltk.org/>, March 2013.

J. O'Shea, Z. Bandar, K. Crockett, and D. McLean: "Pilot short text semantic similarity benchmark dataset: Full listing and description," <http://docm.mmu.ac.uk>, 2008.

J. Oliva, J. I. Serrano, M.D. Del Castillo and A. Iglesias: "SyMSS: A syntax-based measure for short-text semantic similarity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 70(4), pp. 390-405, 2011.

"OMIOTIS: on-line implementation," <http://omiotis.hua.gr>, Aug 2011.

D. M. Pearce, Z. Bandar, and D. McLean: "Using properties to compare both words and clauses," *Proceedings of the 5th KES international conference on Agent and multiagent systems: technologies and applications (KES-AMSTA'11)*, Manchester, UK, pp. 534-543, 2011.

C. Periñán Pascual, and F. Arcas Tunez: "The architecture of FunGramKB", *Proceedings of 7th International Conference on Language Resources and Evaluation*, Malta, pp. 2667-2674, 2010a.

C. Periñán Pascual, and F. Arcas Tunez: "Ontological commitments in FunGramKB". *Procesamiento del Lenguaje Natural*, vol. 44, pp. 27-34, 2010.

L. Qiu, M. Y. Kan, and T. S. Chua: "Paraphrase recognition via dissimilarity significance classification," *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia, pp. 18-26, 2006.

- R. Quirk: *The Use of English*, 2 ed. Longman, London, 1962.
- P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," *International Joint Conference on Artificial Intelligence*, Montreal, Canada, vol. 14(1), pp. 448-453, 1995.
- H. Rubenstein and J. B. Goodenough: "Contextual correlates of synonymy", *Comm. ACM*, vol. 8, pp. 627-633, 1965.
- V. Rus, P. M. McCarthy, M. C. Lintean, D. S. McNamara, and A. C. Graesser: "Paraphrase identification with lexico-syntactic graph subsumption," *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, USA, pp. 201-206, 2008.
- J. Sinclair (ed.): *Collins Co-build Dictionary for Advanced Learners*, 3rd ed. Harper Collins, London, 2001.
- M. Snover, N. Madnani, B. Dorr, and R. Schwartz: "TER-plus: paraphrase, semantic, and alignment enhancements to translation edit rate," *Machine Translation*, vol. 23(2), pp. 117–127, 2009.
- R. Socher, E. H. Huang, J. Pennin, A. Y. Ng, and C. D. Manning: "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," *Advances in Neural Information Processing Systems*, Granada, Spain, 24, pp. 801–809, 2011.
- D. Spoustová, J. Hajic, J. Raab, and M. Spousta: "Semi-supervised training for the averaged perceptron POS tagger," *Proceedings of 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, pp. 763-771, 2009.
- B. Strang: *Modern English Structure*, William Clowes and Sons Limited, London, 1962.
- C. Taggart and J.A. Wines: *My Grammar and I (or should that be 'Me'?)*. Michael O'Mara Books Limited, London, 2008.

K. Toutanova and C. D. Manning: "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, Hong Kong, pp. 63-70, 2000.

G. Tsatsaronis, I. Varlamis and M. Vazirgiannis: "Text relatedness based on a word thesaurus," *Journal of Artificial Intelligence Research*, vol. 37, pp. 139, 2010.

Y. Tsuruoka, Y. Tateishi, J. Kim, Y. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii: "Developing a Robust Part-of-Speech Tagger for Biomedical Text, Advances in Informatics," *10th Panhellenic Conference on Informatics*, Volos, Greece, pp. 382-392, 2005.

A. M. Turing: *Intelligent machines*, Ince, DC (ed.), 1992.

P. D. Turney: "Mining the web for synonyms: PMI-IR versus LSA on TOEF," *12th European Conference on Machine Learning*, Freiburg, Germany, pp. 491-502, 2001.

Z. Ul-Qayyum and W. Altaf: "Paraphrase identification using semantic heuristic features," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4(22): pp. 4894-4904, 2012.

S. Wan, M. Dras, R. Dale, and C. Paris: "Using dependency-based features to take the 'para-farce' out of paraphrase," *Proceedings of the Australasian Language Technology Workshop (ALTW 2006)*, Sydney, Australia, pp. 131-138, 2006.

"WordNet", documentation and implementation, <http://wordnet.princeton.edu/>, 2009.

Z. Wu and M. Palmer: "Verb Semantics and Lexical Selection," *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Las Cruces, NM, USA, pp. 133-138, 1994.

T. Zesch and J. Gurevych: "Analysis of the Wikipedia category graph for NLP applications," *Proceedings of the TextGraphs-2 Association for Comp. Linguistics*, Rochester, NY, USA, pp. 1-8, 2007.

T. Zesch and J. Gurevych: "Wisdom of crowds versus wisdom of linguists - measuring the semantic relatedness of words", *Natural Language Engineering*, vol. 16(01), pp. 25- 59, 2010.

Y. Zhang and J. Patrick: "Paraphrase identification by text canonicalization," *Proceedings of the Australasian Language Technology Workshop*, Sydney, Australia, pp. 160-166, 2005.

S. Zhongzhi: *Advanced Artificial Intelligence*, Beijing: Science Press, Beijing, 2006.

R. Zwaan, "Situation Models: The Mental Leap into Imagined Worlds," *Current Directions in Psychological Science*, vol. 8, pp. 15-18, 1999.

Glossary

The following briefly describes terms and acronyms are commonly used throughout the thesis.

Absolute similarity	The theoretical ideal value of a sentence pair's semantic similarity without further context which may differ from the human scores.
Combiner	A module that combines the similarity from the algorithm using possible rules to produce the final output.
Disambiguation weight	A weight from 0 to 1 to indicate the intended meaning
DTW	Sentence similarity model with dynamic time warping (Liu et al., 2007).
ESA	Corpus based sentence similarity model using Wikipedia (Gabrilovich and Markovitch, 2007).
Framework	The main Linguistic modular framework defined in chapter 4.
Form	How a word is written or spelt.
IISIS	Islam and Inkpen's (2008) sentence similarity model using corpus including word order.
Importance weight	A weight added to a word or clause to indicate its contribution to the overall meaning.
LCH	The lowest common hypernym between two words.
LSA	Latent Semantic Analysis corpus based sentence similarity model (Deerwester et al., 1990).
MSRP	Microsoft Research Paraphrase dataset comprising 1725 sentence pairs in the test set from news sources (Dolan et al. 2004).

OMIOTIS	An ontology based sentence similarity model using all of WordNet connections (Tsataronis et al., 2010).
PCF	Pearson's correlation function.
POS	Part of speech
PoW	Properties of Words (model and meaning architecture using properties.)
PSVO	An order of clauses in the main clause: prepositional, subject, verb and object clauses.
RMS	Root mean square error.
SANO	The final evolution of SARUMAN that extends SCAWIT to handle opposites.
SARUMAN	The main implementation of the sentence similarity from chapter 6 and its extended versions.
SCAWIT	A sophisticated version of SARUMAN that includes clause disambiguation and combining terms.
SRC	Spearman's Rank correlation.
Stemmer	An application to obtain the stem word of a form of a word.
SVO	Subject – Verb – Object, the standard order of clauses in English.
Tagger	An application that adds a linked piece of information such as a part of speech to a word.
Type	The part of speech of a word or a value signifying a similar category to part of speech.
WMS	Word meaning similarity module.
WordNet	(Feldbaum (ed.), 1998) ontological database used as the knowledge base for SARUMAN.

Appendix: Related Refereed Publications

This appendix includes details of the refereed publications to date that relate to this thesis with a copy of each paper attached.

A. Aldeeb, D. M. Pearce, K. Crockett and M. Stanton, "Sentence Similarity Measures to Support Workflow Exception Handling", *Proceedings 12th International Conference on Enterprise Information Systems*, Madeira, Portugal, pp. 256-263, 2010.

D. M. Pearce, Z. Bandar, and D. McLean: "Using properties to compare both words and clauses," *Proceedings of the 5th KES international conference on Agent and multiagent systems: technologies and applications (KES-AMSTA'11)*, Manchester, UK, pp. 534- 543, 2011.