The Best Explanation

Beyond Right and Wrong in Question Answering

Johannsen, Anders Trærup

Publication date:
2013

Document version
Early version, also known as pre-print

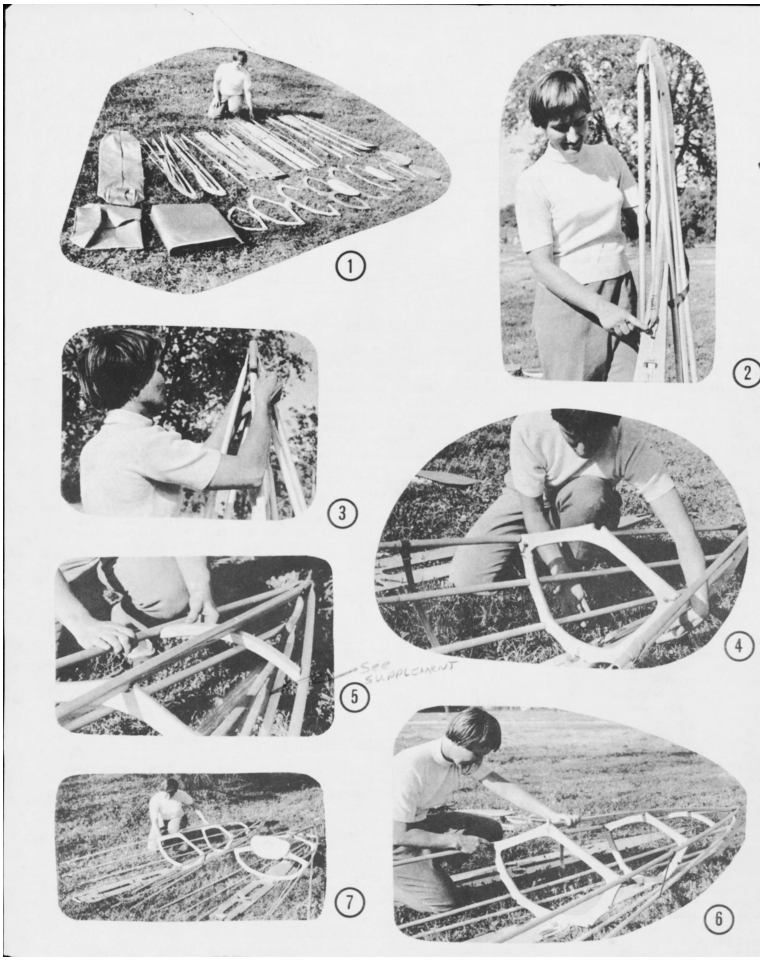
Citation for published version (APA):
Johannsen, A. T. (2013). *The Best Explanation: Beyond Right and Wrong in Question Answering*. Det Humanistiske Fakultet, Københavns Universitet.

ANDERS JOHANNSEN

THE BEST EXPLANATION

THE BEST EXPLANATION

ANDERS JOHANNSEN



Beyond right and wrong in question answering

October 2013

Anders Johannsen: *The best explanation,* Beyond right and wrong in question answering, October 2013

SUPERVISOR:
Associate professor Anders Søgaard, PhD

AFFILIATION:
Centre for Language Technology, Faculty of Humanities, University of Copenhagen

FUNDING:
This thesis was funded by the Experience-oriented Sharing of health knowledge via Information and Communication Technology Consortium (ESICT) grant under the Danish Council for Strategic Research

SUBMITTED:
September 30, 2013

ABSTRACT

There are right and wrong answers, but there are also ways of answering questions which are helpful and some which are not, even when they convey the same information. In this thesis we present a data-driven approach to automatically recognizing the good answers.

A good answer lays out its information so that it is easy to read and understand. And answer structure matters—imagine assembling a piece of IKEA furniture if the order of instructions is scrambled. In general, text is structured by discourse relations, and discourse markers (DMs, e.g. *however*, *moreover*, *then*) are the most apparent and reliable signs of this structure. In the thesis we use DMs to model aspects of answer structure.

Unfortunately, standard discourse processing software make two unrealistic assumptions about DMs, making it hard to apply to community-generated data. They are a) that gold-standard annotations are available for feature generation; and b) that DMs form a closed class for which we have labeled examples of all members. We challenge those assumptions, showing that a) in the absence of gold annotations, state-of-the-art performance can be obtained with much simpler features; and b) sharing features between DMs based on similarity via word embeddings gives an error reduction of at least 20% on unknown DMs, compared to no sharing or sharing by part-of-speech.

Structure-building expressions are often more complex than the simple DMs discussed above and could even be discontinuous (e.g *not only X but also Y*). However, discovering such patterns automatically is a very hard search problem. As an alternative, we generate representations based on regular expressions using data elicited from workers on Mechanical Turk. Using these complex expressions for answer ranking gives an error reduction of 24% compared to a bag-of-words model.

We introduce the task of ranking answers across domains, learning from questions and answers collected from community Q&A sites (cQA). In one experiment we show that importance sampling, where training data is sampled according to similarity between questions, leads to significant improvements over an uninformed sampling strategy.

RESUMÉ

Svar kan være rigtige og forkerte, men de kan også være formulerede på en måde, som gør dem enten nyttige eller ubrugelige, selv hvis de indeholder samme oplysninger. I denne afhandling præsenterer vi en datadreven metode til automatisk genkendelse af de gode svar.

Et godt svar tilrettelægger sit indhold på en måde, der er let at læse og forstå. Svarstruktur er afgørende—tænk på en IKEA-samlevejledning, hvor instruktionerne er blevet byttet rundt. Overordnet set er tekst struktureret af diskursrelationer, og diskursmarkører (DM'er, fx *dog*, *desuden*, *så*) er de mest iøjefaldende og pålidelige udtryk for denne struktur. I denne afhandling bruger vi DM'er som en model for nogle aspekter af svarstruktur.

Uheldigvis gør standardsoftware til at behandle diskurs to urealistiske antagelser om DM'er, hvilket gør det svært at anvende softwaren på almindelig svartekst. Disse antagelser er a) at der findes guldstandardannoteringer til brug for feature-generering; og b) at DM'er udgør en lukket klasse, hvor vi har adgang til opmærkede eksempler af alle medlemmer. Vi anfægter disse antagelser og viser a) at i fravær af guld-annoteringer kan state-of-the-art-niveau opnås med langt simplere features; og b) at feature-deling mellem DM'er baseret på ordindlejringslighed giver en fejlreducering på mindst 20% for usete DM'er i forhold til ingen deling eller deling baseret på ordklasse.

Udtryk, der bygger struktur i en tekst, er ofte mere komplekse end de simple DM'er diskuteret ovenfor og kan endda være diskontinuerte (fx *ikke kun X men også Y*). At finde sådanne mønstre automatisk er et meget vanskeligt søgeproblem. Som et alternativ genererer vi repræsentationer baseret på regulære udtryk ved hjælp af data indsamlet fra Mechanical Turk-arbejdere. Det giver en fejlreduktion på 24% at bruge disse repræsentationer sammenlignet med en bag-of-words-model.

I afhandlingen introducerer vi en ny opgave, nemlig at rangordne svar på tværs af indholdsdomæner. Vores system lærer af spørgsmål og svar indsamlet på brugerdrevne Q&A-sider. Et vigtigt resultat er, at *importance sampling*, hvor træningsdata udvælges på baggrund af ligheden mellem spørgsmål, fører til signifikante forbedringer over en uinformeret strategi for udvælgelse af træningsdata.

PUBLICATIONS

This thesis collects articles written during my time as a PhD student at the University of Copenhagen. The articles have been reformatted for inclusion in the thesis but are otherwise identical to the published versions.

The first set of articles make up the core of the thesis.

- Anders Johannsen and Anders Søgaard. "Learning to disambiguate unknown discourse markers." *Manuscript submitted for publication*, 2013.
- Anders Johannsen and Anders Søgaard. "Disambiguating explicit discourse connectives without oracles." In *IJCNLP*, 2013.
- Anders Johannsen and Anders Søgaard. "Cross-domain ranking of answers using importance sampling." In *IJCNLP*, 2013.
- Anders Søgaard, Héctor Martínez Alonso, Jakob Elming and Anders Johannsen. "Using crowdsourcing to get representations based on regular expressions." In *EMNLP*, 2013.

The second set of articles are related in topic and methods.

- Johannsen, Anders, Héctor Martínez Alonso, Sigrid Klerke, and Anders Søgaard. "Is frequency all there is to simplicity?" In **Sem, ACL*, 2012.
- Anders Johannsen, Héctor Martínez Alonso, Christian Rishøj and Anders Søgaard. "Frustratingly Hard Compositionality Prediction." In *DiSCo, ACL*, 2011.
- Anders Søgaard, and Anders Johannsen. "Robust learning in random subspaces: equipping NLP for OOV effects." In *COLING*, 2012.
- Jakob Elming, Anders Johannsen, Sigrid Klerke, Emanuele Lapponi, Héctor Martínez Alonso and Anders Søgaard. "Down-stream effects of tree-to-dependency conversions." In *NAACL-HLT*, 2013.

ACKNOWLEDGMENTS

A few years ago I took a chair to the blackboard in my office and, on the upper part which cannot be reached from the floor, I wrote: "Nobody reads PhD theses"[1].

This felt reassuring and disturbing at the same time. It told me not to worry too much about results as I spent days researching details about probability theory and similar things which seemed important but were unlikely to make it into a paper. It also had the less appealing message that writing a thesis is a lonely undertaking and, in the end, no one is going to care.

Recently, however, I was compelled to change it. It now says "Someone is reading your thesis," and it is true, even as I write this. In reality, I have benefited a lot from help and supportive environments all along.

First, I would like to thank my supervisor, Anders Søgaard, for always keeping an open door and for being lavish with ideas and concise in critique.

At the CST, I am fortunate to be part of the EMNLP@CPH research group, headed by Anders Søgaard. I wish to thank curent and former group members: Hèctor Martinez, Sigrid Klerke, Jacob Elming, Natalie Schluter, Julie Wulff, Niklas Nisbeth, Christian Rishøj, Barbara Plank, and Dirk Hovy. They have suffered through many drafts of papers and practice talks for conferences, always offering helpful feedback.

I spent six months at Macquarie University, Sydney. I was warmly welcomed by Diego Molla-Aliod, who hosted my stay. I am also grateful to Mark Johnson and his group for letting me be a part of the reading group and activities. The members of the "former reading group" (which had, in fact, not been a reading group for a long time) was another crowd of people whose company I really enjoyed—they assembled Monday evenings at The Ranch. Among these, Francois Lareau, Mark Dras, and Benjamin Börschinger deserve special mention.

In 2012, I was on a three month stay at the University of Edinburgh. Bonnie Webber was kind enough to host me at the Informatics Forum. I am grateful for her patience, and I learned much from our weekly meetings.

1 This, apparently, is something of a tradition among PhD students.

My PhD position was funded by the ESICT project, on a grant from the Danish Council for Strategic Research, and I am thankful for this opportunity. I also wish to thank my colleagues in the project, in particular Jürgen Wedekind, Bente Maegaard, Lina Henriksen, Anna Braasch, and Bart Jongejan.

In all this, my family has been exceptionally supportive of the project, and also quite forgiving. I owe thanks to my sister Rikke, mother Ulla, and my father, Carl Gustav.

Most of all, I want to thank Sigrid to whom I owe the rest.

CONTENTS

1 INTRODUCTION 1
- 1.1 From factoids to fitting answers 1
- 1.2 The recipe for a good answer 2
- 1.3 Learning in the wild 5
- 1.4 Structure of the thesis 6
- 1.5 Contributions of the thesis 8

I TOWARDS BETTER ANSWERS 11

2 A MOTIVATING EXPERIMENT 13
- 2.1 Modeling question type given answers 13
- 2.2 Experimental setup 13
- 2.3 Results 15
- 2.4 Discussion 15
- 2.5 Conclusion 18

3 TYPES OF QUESTIONS 19
- 3.1 What a question is 19
- 3.2 Titles and questions in cQA 21

4 ANSWER QUALITY 23
- 4.1 Quality assessments 23
- 4.2 Conversational and informational questions 24
- 4.3 Caveats in community data 24
- 4.4 Text quality at large 25

5 QUERY-FOCUSED SUMMARIZATION 27
- 5.1 QA and summarization 27
- 5.2 From generic to query-focused summarization 27
- 5.3 Coherence in extractive summaries 28
- 5.4 Discourse-based approaches 29

6 GOOD ANSWER, LONG ANSWER 33
- 6.1 Introduction 33
- 6.2 Experiments 36
- 6.3 Results 38
- 6.4 Human evaluation 39
- 6.5 Conclusion 40

II ARTICLES 41

7 CROSS-DOMAIN ANSWER RANKING USING IMPORTANCE SAMPLING 43
- 7.1 Introduction 43
- 7.2 The StackQA corpus 45

7.3 Feature sets 45
7.4 Importance sampling 46
7.5 Experiments 47
7.6 Results 47
7.7 Discussion 49
7.8 Related work 50
7.9 Conclusion 51
8 USING CROWDSOURCING TO GET REPRESENTATIONS BASED ON REGULAR EXPRESSIONS 53
8.1 Introduction 53
8.2 Experiments 54
8.3 Results 60
8.4 Related work 61
8.5 Conclusion 61
9 LEARNING TO DISAMBIGUATE UNKNOWN DISCOURSE MARKERS 63
9.1 Introduction 63
9.2 Transfer learning 64
9.3 Discourse markers 67
9.4 Disambiguation experiment 70
9.5 Sense classification experiment 73
9.6 Results 75
9.7 Discussion 76
9.8 Related work 76
9.9 Conclusion 78
10 DISAMBIGUATING DISCOURSE CONNECTIVES WITHOUT ORACLES 79
10.1 Introduction 79
10.2 The importance of the long tail 81
10.3 Experiments 82
10.4 Results 84
10.5 Discussion 84
10.6 Related work 86
10.7 Conclusion 86
11 DOWN-STREAM EFFECTS OF TREE-TO-DEPENDENCY CONVERSIONS 89
11.1 Introduction 89
11.2 Applications 93
11.3 Results and discussion 100
11.4 Conclusions 103
12 ROBUST LEARNING IN RANDOM SUBSPACES: EQUIPPING NLP FOR OOV EFFECTS 105
12.1 Introduction 105

12.2 Robust learning under random subspaces 107
12.3 Evaluation 110
12.4 Related work 114
12.5 Conclusion 116
13 FRUSTRATINGLY HARD COMPOSITIONALITY PREDICTION 117
13.1 Introduction 117
13.2 Features 118
13.3 Correlations 121
13.4 Regression experiments 121
13.5 Discussion 123
13.6 Conclusions 124
14 IS FREQUENCY ALL THERE IS TO SIMPLICITY? 127
14.1 Introduction 127
14.2 Features 127
14.3 Unlabeled data 130
14.4 Ranking 130
14.5 Experiments 132
14.6 Discussion 133
15 CONCLUSION 135

III APPENDIX 137
A SAMPLE OF CQA TITLES 139
B A CASE STUDY OF A SUMMARIZER 143
C FEATURES TRACKING LENGTH 149

BIBLIOGRAPHY 151

LIST OF FIGURES

Figure 1 Structure of a question page 14
Figure 2 Average answer length, time rank and score rank 35
Figure 3 Length of top ranked answers and percentage of answers reaching top score rank 36
Figure 4 Results selecting N features using χ^2 58
Figure 5 Feature combination techniques 59
Figure 6 Histogram of correlation between POS of previous token and label 66
Figure 7 Distribution of DMs 80
Figure 8 Clear and difficult cases in dependency annotation 90
Figure 9 Head decisions in conversions 91
Figure 10 CoNLL 2007 and LTH dependency conversions 91
Figure 11 Features used to train CRF models 95
Figure 12 Examples of SMT output 99
Figure 13 Sentence compression output 99
Figure 14 Distribution of dependency labels 102
Figure 15 Optimal decision boundary 107
Figure 16 Robust learning in random subspaces 110
Figure 17 Hierachical structure of 20 Newsgroups 111
Figure 18 Plots of P-RLRS error reductions 113
Figure 19 Classifier comparison 115
Figure 20 Removal rates sampling ξ 115
Figure 21 Correlation coefficients 122
Figure 22 Subjective judgements about markedness 124
Figure 23 Kappa score vs. number of data points 134
Figure 24 Example of solutions y 146
Figure 25 Normalized value vs. document length 150

LIST OF TABLES

Table 1 Examples of argumentative phrases 5
Table 2 Performance of *how* vs. *why* classification 15
Table 3 Top features in *how* vs. *why* classification 16
Table 4 Percentage classified as DM 17
Table 5 Question title types 21
Table 6 Dimensions of answer quality 24
Table 7 Results for essays 38
Table 8 Ranking of StackOverflow answers 39
Table 9 Observed agreement 40
Table 10 Most similar out-of-domain questions 48
Table 11 Feature ablation 48
Table 12 Ranking performance 49
Table 13 Characteristics of the feature sets 54
Table 14 Results using all features 57
Table 15 DM disambiguation results 75
Table 16 DM sense classification results 75
Table 17 Flat DM clusters 77
Table 18 Score on oracle and predicted features 84
Table 19 F_1 score per connective 85
Table 20 Down-stream performance results 97
Table 21 Results on 20 Newsgroups 112
Table 22 Results on the EWT 114
Table 23 Highest correlated features 128
Table 24 Performance on POS 133

1 INTRODUCTION

1.1 FROM FACTOIDS TO FITTING ANSWERS

Philosophers make a distinction between knowing *why* or *that* something is the case, and knowing *how* to do it—between knowing-why and knowing-how. For some philosophers, this distinction is at the heart of things. Knowing-why is fundamentally different from knowing-how; it is a separate mode of knowledge. Evidence from cognitive science seems to support this—*procedural knowledge* and *propositional knowledge* are encoded differently in the brain.

Asking questions is one way to expand our knowledge of the world. To be concrete, say you are walking in your garden and suddenly encounter a swarm of bees hanging from a branch of a tree. Unless you have seen this kind of thing before, you are probably wondering *what to do*. Consider then the two pieces of information in Ex. (1.1) and Ex. (1.2).

(1.1) *Why do bees swarm?* Honey bees swarm because they are looking for a new site to form a new colony. It is a natural and positive means of population increase[1].

(1.2) *How do I handle a swarm of bees?* If you have a swarm of honey bees which have suddenly arrived, please do not panic and rush out to kill them. There are plenty of beekeepers available who are more than willing to come out and collect your swarm[2].

The examples are answers to two distinct, although related, questions. There is nothing wrong with the information in Ex. (1.1); it begins to explain the phenomenon but fails to mention what to do about it. In the situation at hand, it answers the wrong question: the *why* instead of the *how*. In contrast, Ex. (1.2) seems a better choice, since it addresses the situation at hand and directly instructs us what to do (and what not to do).

1 http://www.wildlifeextra.com/go/uk/bee-swarm.html
2 http://www.wasp-removal.com/bee-removal.php

In the above examples the appropriateness of the answers can be traced back to the presence or absence of certain linguistic traits. The answer to the *why* question explains causal relationships and uses the discourse marker "because" to connect the cause and the effect. The *how* answer uses conditionals (if X, then Y) to ensure that the actionable advice it gives fits the situation. Additionally, it speaks to the reader directly using second-person pronouns and emphasizes its advice with "please".

More generally, *how*, *why*, and other types of questions set up different expectations about the linguistic structure of a good and fitting answer. In the thesis we explore how these expectations can be modeled empirically and used to identify good answers.

1.2 THE RECIPE FOR A GOOD ANSWER

The idea, then, is that good answers differ from answers of lower quality in terms of how they structure their content, and that this structure is conditional on the type of question. Setting aside the issue of answer correctness[3], we ask: Can such a recipe for good answers be learned? We address this by asking several more questions, discussed below:

1. Where do we find corpora of good and bad answers?
2. What are the biases of those corpora?
3. What is a good representation of an answer?

GOOD AND BAD ANSWERS

At first sight community-based Q&A sites (cQA) offer the ideal platform for investigating the issue of good and bad answers, for several reasons. A question posed to one of these sites usually attracts answers by several people and the answers are ranked by community voting, so that for each question a prioritized list of answers can be extracted. By comparing higher and lower ranked answers to the same question we can check for systematic differences and describe what they are. Furthermore, the data is abundant; as of September 2013, the Stack

3 A brief discussion is found in Chapter 7

Exchange network of cQA sites, which we make use of in Chapter 7, hosts 7.1 million questions and 12.8 million answers distributed on 106 sites with distinct profiles and subjects[4]. Yahoo! Answers, a general-purpose cQA site founded in 2005, remains the largest resource and exceeds the combined figure by several orders of magnitude. In 2010, the site thus passed the one billion mark for answers[5].

The article "Cross-domain answer ranking using importance sampling" (Chapter 7) attempts to reproduce the answer rankings of 30 Q&A sites on a variety of subjects, all belonging to the Stack Exchange network. While answer ranking is not a new task [161], we reframe it by introducing one important restriction, which is that features should derive from the textual content of question and answers and not rely on the social structure of the cQA site. This effectively widens the scope of application for the model considerably beyond cQA, even though it is learned from cQA data. It could, for instance, be used to find answers embedded in running text on web pages or to guide content selection in query-focused summarization (a possibility we return to in Section 5).

BIAS IN COMMUNITY-GENERATED DATA

Unfortunately, the ratings awarded to answers in community voting are not the transparent proxies for answer quality that we might like them to be. People submit votes and choose favourite answers for a number of reasons, some of which are unrelated to answer quality, such as feeling gratitude (see Section 4). More importantly, the social practices around question answering, including the reward systems of cQA sites, strongly influence how votes are cast [6]. One finding of Chapter 6 is that the timing of an answer (does it arrive first, second, etc.), a factor seemingly unrelated to quality, nevertheless correlates with its rating and length.

ANSWER REPRESENTATION

Once we have a corpus of ranked answers, we need to find a representation, which

4 http://stackexchange.com/

5 http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served/

1. captures (aspects of) the structure of the answer, and
2. can be reliably produced on large amounts of data.

The first requirement states that the representation should capture facts about how the answer organizes its information, i.e. how the pieces of the answer relate to each other. The second requirement, which is dictated by our choice of corpus, makes it necessary to go for a relatively shallow approach. Together these requirements point towards the use of discourse markers (DMs). DMs explicitly signal the presence of discourse relations, which help structure text and make it coherent. A discourse relation holds between two units of text, assigning a "meaning" or sense to that relation. For instance, the relation between the bracketed expressions in Ex. (1.3) is one of CONTRAST.

(1.3) [Sam goes to work], **but** [Jim stays at home].

While DMs are not the only way of establishing discourse relations — in fact, many discourse relations are established through simple adjacency of sentences — they can be detected more reliably than other types of discourse relations and are thus the logical point of departure.

Our use of disambiguated and sense-assigned DMs as a representation of answer structure was initially met with problems, as the publicly available software for performing disambiguation and sense classification[6] did not perform as well as expected on our data. This prompted further investigations into the task and resulted in the two papers 'Disambiguating discourse connectives without oracles" (Chapter 10) and "Learning to disambiguate unknown discourse markers" (Chapter 9).

In a complimentary line of work we look at deriving representations based on regular expressions. For a motivation, consider Table 1, which showcases some examples of useful phrases for writers of argumentative essays [37]. Intuitively, phrases like these seem useful in capturing the structure of an answer. However, they would be difficult to discover automatically, because although they are shallow patterns, some are disjoint and some are only semi-fixed expressions that allow for some variation in syntax or word choice. In the article "Using crowdsourcing to get representations based on regular expressions" (Chapter 8), we study an alternative strategy of asking people to supply the patterns.

6 We used the Logistic Regression classifier of [136]

Expanding	neither ...nor; not only ...but also; it is found that ...
Examples	which is to say; in other words; for example; as in the following examples; such as; in particular
Quotes	according to X; to quote from X; X (tell | show | argue) that; X state; X discuss; X express the concern
Uncertainty	possibly; perhaps; very likely; it might; unsure
Returning to emphasize	even if X is true; although X may have a good point; all the same; in spite of X

Table 1: Examples of argumentative phrases from Cottrell [37].

1.3 learning in the wild

An orthogonal view on the papers of this thesis is that together they make a statement about the proper evaluation of NLP tasks. They insist it should be realistic with respect to the expected performance in the wild. Evaluation may not reflect real-life performance for a number of reasons, including

1. loss due to domain differences between training and test data;
2. use of privileged information (e.g. gold annotations) not available at test time (in the wild);
3. use of an improper metric for the task; and
4. only doing intrinsic evaluation, especially for tasks like parsing and tagging, which are often components of larger NLP systems.

In "Disambiguating discourse connectives without oracles" (Chapter 10), we argue that published results for DM disambiguation partly give the wrong impression, because they use averaging over instances (allowing a few high-frequency items to dominate) instead of averaging over all types of DMs. When the latter method is adopted, results are much less impressive, suggesting that more work is needed still.

In "Learning to disambiguate unknown discourse markers" (Chapter 9), we challenge the assumption that labeled training

material is available for all DMs (and provide evidence to the contrary). The paper then goes on to suggest an evaluation setting where classifiers are trained specifically for each DM, on training data which excludes any instances of that marker.

Ideally, evaluation of a task should not be done in isolation but involve measuring the performance "down-stream", in applications that rely on the output of the task. For instance, summarization often requires part-of-speech tagged input, and sentence compression commonly operates on the syntactical structure produced by a parser. The relation between intrinsic and down-stream performance is by no means trivial. As we demonstrate in "Down-stream effects of tree-to-dependency conversions" (Chapter 11), what appears to be a relatively innocent choice of constituency-to-dependency-parsing conversion scheme has dramatic effects on down-stream performance.

More often than not, the data used for training and evaluating a model looks very different from the data it is going to be applied to. In data-driven approaches standard procedure is to first partition data into disjoint training and test sets and then induce a model from the training set. The performance of the model on the test set measures the generalization capability of the model, but only given certain assumptions, the most important of which is that training and test data are both sampled independently and identically distributed (*i.i.d.*) from the underlying "true" distribution. In reality, the standard annotated corpora in NLP, such as Penn Treebank and PDTB, are highly biased samples (of the distribution of English text). This is unfortunate in at least two respects. Not only does it make performance figures on the test set less meaningful as a measure of generalization ability, it also makes us vulnerable to out-of-vocabulary (OOV) effects, which happen when features that were present in the training data go missing at test time and vice-versa.

We propose a method for dealing with OOV errors in "Robust learning in random subspaces: equipping NLP for OOV effects" (Chapter 12)

1.4 Structure of the thesis

The first part of the thesis looks at the components of a good answer. In a motivating experiment, Chapter 2 lends support to the idea that particular types of questions produce particular structures in the answers. Chapter 3 takes a step back, ask-

ing what a question really is. It turns out that not all questions look like questions in syntax, which leads us to categorize alternative forms of questioning in community-generated data. Chapter 4 addresses another fundamental issue, the definition and evaluation of answer quality. Chapter 5 opens with the observation that summarization and complex question-answering have similar aims and could benefit from sharing approaches. We then propose specific ways of altering summarization systems to further our goal of getting better answers. Chapter 6, the final chapter of the first part, brings together many of the themes discussed so far. In the chapter we rank answers from a cQA site using discourse-related features.

The second part consists of published[7] articles, written from 2011 to 2013. Chapter 7 revisits the ranking problem of Chapter 6, casting it as a problem of domain adaption, using data from 30 cQA sites. Although results improve over a baseline, much of the variance in the data is not accounted for by the model. Hypothesizing that our text features are not sufficiently expressive, we consider, in Chapter 8, how to elicit discontinuous, regular expression-like features from experts and by use of crowd-sourcing.

Another cause of the unexplained variance may be that the discourse processing is not working as advertised on community-generated data. Chapter 9 and Chapter 10 deal with recognizing discourse markers and classifying the sense of discourse relations. Both chapters point out ways in which standard evaluation is not realistic and therefore fails to promote building robust models. We suggest new evaluation methods, which, importantly, lead to new approaches.

Throughout Chapter 9 and Chapter 10, we emphasize the need for realistic evaluation. This is also the motivation behind Chapter 11, where we investigate the dramatic effect the choice of tree-to-dependency conversion scheme has on down-stream application performance. Chapter 12 proposes a solution to a common problem in NLP, out-of-vocabulary errors due to a domain mismatch between training and test data.

The final chapters, Chapter 13 and Chapter 14, are about lexical choice. Both describe systems competing in shared tasks. (The system in Chapter 13 won). In Chapter 14 we describe a system ranking lexical substitutions according to perceived simplicity. Ensuring reader-appropriate vocabulary is integral to text quality (see Chapter 4) and thus lies on the path towards

7 With the exception of Chapter 9, which is in peer-review.

better answers. The problem of Chapter 13 is to recognize non-compositional expressions (e.g. that a "hot dog" is not a hot *dog*). In the context of lexical choice, compositionality detection enables us to identify which words can be meaningfully substituted.

We conclude in Chapter 15.

1.5 CONTRIBUTIONS OF THE THESIS

Below, we highlight the main contributions of the thesis, which are made in the areas of discourse processing, answer ranking, and answer representation.

Discourse processing

We challenge two assumptions about labeled data commonly made in discourse processing:

1. it is feasible to have labeled examples of all discourse markers (Chapter 9); and
2. hand-annotated data (e.g. gold parse trees) is available for constructing features (Chapter 10).

Adopting new evaluation procedures that get rid of these assumptions, we show that

1. sharing features between discourse markers based on syntactical evidence from word embeddings yields an error reduction of at least 20%, compared to no clustering or clustering by part-of-speech (Chapter 9); and
2. models based on simpler and more robust features perform at the same level as state of the art (Chapter 10).

Answer ranking

We introduce the task of ranking answers across domains using cQA data and demonstrate that a variation of importance sampling, where training data is sampled according to similarity between questions, leads to significant improvements over randomly sampled training data (Chapter 7).

Analyzing community-generated quality assessments, we show that major biases exist with respect to

1. answer time; and
2. answer length.

We propose a sampling method to deal with answer-time bias but report a negative result concerning answer-length bias (Chapter 6).

Answer representation

We describe a method of generating discontinuous, regular expression-like representations for text classification tasks, such as answer ranking. We use crowdsourcing as a way of avoiding the astronomical search space of regular expressions. Using these representations for answer ranking, we obtain an error reduction of 24% in comparison with a bag-of-words model (Chapter 8).

Part I

TOWARDS BETTER ANSWERS

2

A MOTIVATING EXPERIMENT

What do we need in order to get better answers? In this chapter we motivate the idea that answer structure depends on question type by doing a small experiment on cQA data.

2.1 MODELING QUESTION TYPE GIVEN ANSWERS

When designing a QA system it is important to make sure that the answer given is appropriate for the question asked. The QA system thus needs to have a model of the answer type given the question. In case of a probabilistic system it means modeling the conditional distribution $P(A|Q)$.

In this experiment we look at the problem from the other side. Given an answer can we predict what kind of question triggered it? — i.e. can we model $P(Q|A)$? Intuitively, if a high accuracy model for $P(Q|A)$ can be learned, it suggests a strong link between question and answer, which in turn emphasizes the need to have an accurate model of $P(A|Q)$ in the QA system.

2.2 EXPERIMENTAL SETUP

We wish to predict the type of question from the text of the answers on the question page. The data for the experiment is a corpus of 94,609 question pages collected at 30 QA sites in the Stack Exchange network[1]. Question pages were sampled at random, allowing a maximum of 5,000 pages from any one site. The elements of a question page is shown in Figure 1. The page is headed by a question and consists of one or more answers. The question is subdivided into a title, displayed in a larger font, and a body for elaborating the question. As a result of this design, the question text may contain several distinct question sentences. For present purposes we define a question sentence as any sentence in the question body or title whose last token is a question mark.

First, we simplify things by considering only two types of questions: a) manner questions that request instructions about

1 The corpus is described in more detail in Chapter 6.

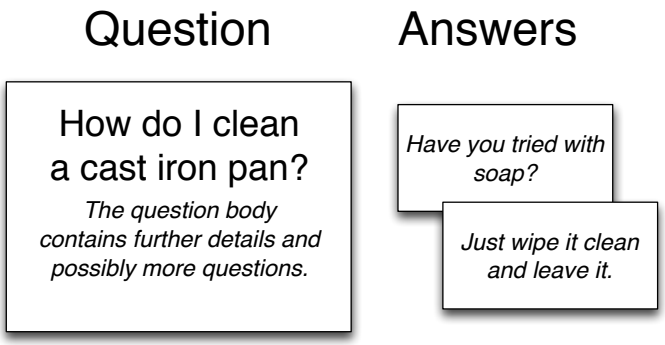


Figure 1: Structure of a question page

how to accomplish something; and b) explanation questions that ask *why* something is the case. Following Surdeanu et al. [162], *how* questions should match the regular expression below:

```
how (to|do|did|does|can|would|could|should)
```

A *why* question only needs to match the simpler regular expression `why`. Examples of both question types are listed as Ex. (2.4) - (2.7).

(2.4) In short, why does G-d allow evil to exist? [*why*]

(2.5) Theoretical: why there's no gradient of doneness in bread? [*why*]

(2.6) how do you avoid the circularity in this argument? [*how*]

(2.7) how to curb the smell of fish? [*how*]

Questions are classified as either *why* or *how* if at least one of the question sentences match the regular expression. If both regular expressions match, or neither match, the whole question page is discarded.

We model the conditional probability of the question type given the answer using logistic regression. Whenever

$$P(Q = \text{why}|A) \geqslant P(Q = \text{how}|A)$$

we predict that the question is *why*, and else *how*.

The experiment compares three feature representation of the answers. The first representation is a bigram bag-of-words model using raw counts and normalizing each feature vector to unit length. The second and third representations record only how

Answer representation	Accuracy
Bag of words	76%
DMs (disambiguated)	65%
DMs (list)	64%
Majority baseline	50%

Table 2: Performance of *how* vs. *why* classification

often a set of discourse markers appear in the answers. Both are restricted to a fixed list of 100 DMs (those annotated in the PDTB [139]), but differ in the following way: one counts only DMs disambiguated using standard software [136], while the other simply matches DMs by string value.

2.3 RESULTS

Table 2 shows the accuracy obtained under the three representations of the answers. The DM representations score 65% for disambiguated markers and 64% for list matching and are thus very close to each other but considerably above the random baseline of 50%. In this task disambiguation with standard tools offers little or no advantage. The bag-of-words model, which has a much richer representation of the answers[2], gives the best accuracy, at 76%.

These results suggest that the feature representations capture meaningful aspects of answer structure.

2.4 DISCUSSION

We now consider what expressions (DMs and bigrams) were effective in discriminating between the two types of questions. Table 3 lists the top 10 features for each representation, ordered by the absolute size of the coefficient in the fitted logistic regression model. In all representations, "because" either tops the list or comes in a close second. As expected, this causal marker is strongly associated with explanation-type questions.

In the bag-of-words model the top features show interesting patterns. First, the words "why" and "how" seem to echo the questions they answer; and while some of these do come in the

2 The list-based DM model is a subset of the bag-of-words model, except for the 12 DMs that consist of three tokens or more.

Bag of words BOW		DMs (DISAMB.)		DMs (LIST)	
why	3.84	thus	2.55	because	5.10
because	3.74	because	2.21	thus	3.35
you can	-3.70	therefore	2.17	therefore	3.06
was	3.55	whereas	2.03	then	-2.14
how	-3.23	hence	1.62	as	1.95
reason	3.17	in other words	1.59	when	1.64
when	2.64	later	1.57	indeed	1.50
code1	-2.41	alternatively	-1.53	until	1.49
you could	-2.33	in fact	1.52	earlier	1.48
way	-2.29	finally	-1.26	unless	-1.47

Table 3: Top features in *how* vs. *why* classification. In this binary classification task *why* was coded as 1 and *how* as 0. Positive coefficients thus contribute towards a classification as *why* and negative coefficients towards *how*.

form of rhetorical questions ("Why is this?"), the main reason is quite simply that answers to manner questions talk more about manner and answers to explanation-type questions are more concerned with reasons. Second, the phrases "you can" and "you could" are strong predictors of *how*. These phrases emphasize the person-to-person mode of communicating in QA by addressing the asker directly (the second person pronoun "you", which is not shown in the table, also associates strongly with *how*) and by suggesting actions. We return to the use of pronouns in cQA in Section 6.2.3 and 7.3. Finally, "was" is highly predictive of *why*, a fact which might seem puzzling since we have no reason to suspect that this function word is differently distributed in the two groups of answers. The explanation is that "was" acts as a proxy for the length of the answers, which, on average, is 283 tokens for *how* and 349 tokens for *why*. Note that this happens despite unit length (L_1) normalization of the feature vectors. See Appendix C for more details.

The coefficients obtained using the DM representations are similar, with a few notable exceptions. 1) Although "because" is a strong feature in both representations, it has a much higher weight in LIST. As can be seen in Table 4, 87% of occurrences of "because" are classified as DMs and consequently 13% are

		DM %	
	N	Disamb.	PDTB
alternatively	168	86%	100%
as	71,722	2%	13%
because	4,407	87%	63%
earlier	183	0%	2%
hence	277	45%	33%
in fact	395	73%	87%
in other words	159	88%	89%
indeed	305	45%	81%
later	672	26%	34%
then	7,244	69%	64%
therefore	642	84%	92%
thus	655	81%	92%
unless	671	91%	94%
until	906	65%	41%
when	7,527	83%	65%
whereas	166	69%	100%

Table 4: Percentage classified as DM. N is the number of DM expressions found by string matching. The disamb. column shows how many of these were classified as DMs by the Pitler and Nenkova [136] software, expressed as a percentage of N. For comparison, the last column gives the percentage of DM expressions annotated as DMs in the PDTB.

not counted in DISAMB. Even if we assume that the classifier is correct in leaving these out, the use of "because" in a non-discourse sense (e.g. "because of") still indicates that the answer is about causes and reasons. 2) While "as" is a top five feature for LIST, it receives little weight in DISAMB. The token appears very frequently in the answers, and in language overall, but only rarely with a discourse function (see Table 4). It is likely that "as" serves the same function of tracking answer length as "was" does in the bag-of-words model[3].

3 See Appendix C.

In this classification experiment, using disambiguated DMs comes with no apparent advantage in terms of higher accuracy in predicting the question type. The payback, however, is that disambiguation avoids some confounding factors in the feature model, for instance having a feature that simply counts occurrences of the very common token "as". We did not discuss what impact the quality of the classifier has on the results; but see Chapter 7 for an evaluation of the classifier in a similar setting. Finally, the superior performance of the bag-of-words model shows that there are systematic differences between the answer groups beyond what is captured by the surface discourse structure of the DM models.

3

TYPES OF QUESTIONS

3.1 WHAT A QUESTION IS

In Section 2 we took any sentence with a question mark as the last token to be a question. While this test is simplistic, automatically identifying questions in free text is in general not easy, because utterances that function as questions—or "do questioning"—are often not marked syntactically [57]. Conversely, sentences which, judged by their syntax, appear to be questions may not actually do any questioning. Ex. (3.8) and (3.9) illustrate both types.

(3.8) Another cup of tea?

(3.9) What's the use?

Questions may be defined by their *syntax*, *semantics*, or on *pragmatic* grounds. We will look at each one in turn.

SYNTAX The prototypical question in English uses a special clause form for questions, the interrogative. It is characterized either by the use of an initial *wh* word and subject-object inversion; or, in case of *yes-no* questions, by placing a function word (e.g. "do") in front of the subject [64]. However, there is no one-to-one correspondence between the syntactical category of interrogative and what is generally recognized as questions, and the non-marked questions are not merely quirky corner-cases. One study of how the non-interrogative form is used in conversations puts the usage figure for the canonical interrogative form at 59%, while the remaining 41% perform questioning with alternative syntactical constructions [181].

SEMANTICS In semantics, a question is any expression that defines a set of logically possible answers [74, p. 866]. Examples (3.10) through (3.12) list sample responses to the question "Have you seen it?":

(3.10) a. No

b. I have

(3.11) a. I'm not sure
b. I can't remember
c. Does it matter?

(3.12) a. I've already told you that I have
b. It's on your desk
c. I saw it yesterday [74]

Only the responses in Ex. (3.10) are in fact answers. The semantic view introduces a distinction between answer and response, and any reaction that falls outside the set of logically possible answers is called a response. The expressions in Ex. (3.11) fail the test, because they do not provide the required information. And while the examples in (3.12) contain information pertaining to the question from which the answer may be inferred, and even add useful details compared to the straight answer, under the semantic view they are not considered answers.

PRAGMATICS Bolinger [21], in an influential work on the direct question in English, takes the view that questions are essentially psychological entities that defy linguistic definition. To be sure, there are prototypical members of the class of questions, but necessary properties are impossible to identify. By this view, a question is any expression which *typically* elicits a response.

The syntactic view of questions is important in NLP, because at present it is more amenable to computational modeling than the pragmatic view and does not suffer from the knowledge acquisition bottleneck of semantics. So while it is probably true that no neat syntactical definition of a question can be found, the consequence is not that we should disregard clues from syntax altogether.

Finally, we bring to mind that better answers, as they are outlined in this chapter, are not answers in the strict sense proposed by the semantic definition; they are responses. It is quite unlikely that a user (or indeed anyone) would know how to phrase questions such that his or her information need would be satisfied by their answers. For better answers the objective must be to deliver contextually helpful responses.

Title type	
Action	31%
Other	21%
Object	18%
Manner	12%
Symptom	10%
Interrogative	8%

Table 5: The percentage of each kind of question title in a random sample (N = 100) of titles that do not end in question marks. Details about the categorization is in Section 3.2.

3.2 TITLES AND QUESTIONS IN CQA

While the discussion above is of a more theoretical nature, we now turn to the behavior of questions in corpus data. We perform a qualitative analysis of non-interrogative questions on cQA sites. Community QA sites commonly require askers to supply a title for their question. We examine a sample of 94,609 such titles from 30 sites in the Stack Exchange network. The data is the same as in Section 2, except here we are only interested the contents of the title field. Stack Exchange directly instructs users to write a question. On StackOverflow, for instance, a help text placed inside the title field asks: "What's your programming question? Be specific." The other sites use a similar phrasing, swapping "programming" for the subject of the site.

Interestingly, only 56% of the titles end in a question mark. The figure varies between 30% and 93% for individual sites. To find out whether the titles without question marks represent genuine non-interrogatives, we extracted 100 such titles at random and looked at them manually[1]. In only 8 cases did we find an interrogative form where the question mark is left out; the rest were various non-interrogative forms. Note that titles with question marks may also be non-interrogatives, making the percentage even higher.

Table 5 gives an overview of the types of non-interrogatives found in the sample. We describe the categorization scheme below, giving examples of each type. The categories were determined by analysis of the data. The data was annotated by the

1 The sample is included as Appendix A

author of the thesis and the results were discussed with a colleague. If a title did not clearly fall within one of the categories, we labeled it as OTHER.

ACTION The action the asker wishes to perform:

(3.13) Execute command on shared account login

(3.14) Grab certain contents of a file

OBJECT The object that the question concerns:

(3.15) Perspective in early pseudo-3d games

(3.16) Electron transitions in an infinite square well

(3.17) arduino 3x3 LED matrix

MANNER The manner in which an action should be performed:

(3.18) how to determine drive times like those available in google maps

(3.19) How to have overlapping under-braces and over-braces

SYMPTOM An error or undesirable state-of-affairs:

(3.20) hard crack candy coming out too sticky

(3.21) incoming mail just sits in the drop folder

Important subtypes of the OTHER category are *hypothetical object* and *comparison*:

(3.22) Toaster Oven pan Without The Toaster Oven

(3.23) "Anxious to" versus "eager to"

A hypothetical object, such as in Ex. (3.22), is some entity or situation which the asker is interested in bringing about. A title with a hypothetical object can often be paraphrased as a question with "Is it possible to have X?" or "How can I accomplish X". In some respects it is the opposite of SYMPTOM which is concerned with an actual but undesirable situation, whereas the hypothetical object is about a future situation being actively pursued. Comparisons, like the one in Ex. (3.23), contrast two options and ask for the most suitable.

4

ANSWER QUALITY

We discuss what goes into answer quality and how to obtain assessments of quality. Community-generated data is biased in various ways, and we examine the consequences of this next. Finally, we broaden the perspective and consider answer quality in terms of general text quality.

4.1 QUALITY ASSESSMENTS

Studies on answer quality in cQA generally rely on quality assessments that are either

1. generated by the *community* as an integral part of cQA, or
2. annotated by *experts* post hoc.

The community assessments come from activities performed on the cQA sites and include up-voting and down-voting, liking, chosing as a favourite, and selecting as "best answer" (only available to the asker). Both sources of quality assessments have advantages and drawbacks. Community assessments are abundant and cheap, whereas an expert study costs money and takes effort to organize. On the other hand, experts can be asked to judge fine-grained quality criteria directly, whereas quality judgements from the community are indirect and must be inferred from user actions. Nonetheless, community-generated data is vital for building data-driven automatic systems, and expert judgements can provide important insight into what this data means.

Studies using experts usually decompose answer quality into a number of dimensions, which are then evaluated on a yes-no [54] or Likert [104] scale. While the dimensions vary in number and granularity between studies, the list in Table 6 is typical [151, 54]. Note that readability and coherence are not mentioned as evaluation criteria, presumably because the answer quality criteria were not developed with natural language processing tasks in mind.

Dimension	Description
Complete	The answer addresses all parts of the questions
Accurate	The answer is correct
Verifiable	The answer references external sources
Timely	The answer arrives quickly
Useful	The answer gives the asker what he or she needs

Table 6: Dimensions of answer quality

4.2 CONVERSATIONAL AND INFORMATIONAL QUESTIONS

Questions are posed for different reasons and this matters because the perceived quality of an answer depends on what the goal of the question is [87, 86]. We will see an example of this in Section 4.3. Now consider Ex. (4.24) and (4.25) below:

(4.24) What is the most effective treatment for bed bugs?

(4.25) What is the best superpower to have?

Both questions are likely to attract several answers, and participants might end up being in sharp disagreement about the right answer. However, in case of Ex. (4.25), objective facts are unlikely to settle the discussion. In fact, the goal of the question appears to be to have the discussion. Where Ex. (4.24) expresses a real information need, Ex. (4.25) is just making conversation. Similarly, Harper et al. [69] characterize questions posted on cQA sites as either *conversational* or *informational* and show that these categories can be annotated with high agreement. They also successfully train a classifier, noting the strongest features are related to pronoun use: the use of "I" points towards a categorization as *informational* while "you" is a strong predictor for *conversational*. A similar study uses machine learning to identify questions that lead to subjective answers [3].

4.3 CAVEATS IN COMMUNITY DATA

Many researchers [152, 161] use the best answer, as chosen by the asker, as a quality measure, but there are several shortcomings to this approach. The best answer status reflects a single subjective opinion from a person who may not be enough of an expert to truly judge whether the answer is good [150].

Kim and Oh [86] show that, on Yahoo! Answers, the best answer is often chosen to display gratitude towards a helpful, or quick, answerer rather than to promote the best answer. Further, choosing a best answer is a voluntary action, which is sometimes not performed, and once the best answer has been selected, it may not be updated in the light of new answers, although it is technically possible to do so.

Kim et al. [87] look into reasons people give for choosing the "best answer" on Yahoo! Answers. Analyzing a sample of 465 comments left by the asker to explain their choice, the authors find that 33% value socio-emotional aspects, for instance that the answerer put in an effort preparing the answer, or that they agree with the answer. The subjectivity involved in "best answer" selection has led some researchers to caution against the use of "best answers" in studies [54]. However, if we break down the numbers by question type, the picture changes somewhat. In opinion-type questions, which are subjective and similar to the conversational questions above, 58% of are selected for socio-emotional reasons, while the same figure for the more objective information questions is only 13% [87].

Most studies analyze answer quality at the level of the answer. However, Anderson et al. [6] argue that individual answers to a question do not compete but are instead complementary to each other, focusing on various aspects of the answer. Therefore, quality should be seen as a property of the question page and all answers in combination. They predict the long-term impact of the question page, measured as the number of page views it has attracted after one year

4.4 TEXT QUALITY AT LARGE

Text quality is hard to pin down as a concept, even though most have no trouble recognizing poor or high quality specimens when they see them. Text quality is something above and beyond *readability* and *linguistic quality*. Linguistic quality concerns grammaticality, referential clarity, focus and coherence, while readability indexes typically are computed on the basis of word complexity and sentence length [119, 88]. A text with low linguistic quality is a bad reading experience and could even be unintelligible. In contrast, a text with low readability is simply difficult. To have high readability, we need high linguistic quality. Likewise, high (or appropriate) readability is necessary for high text quality.

In a recent paper, Louis and Nenkova [105] offer a new direction for the study of text quality. Their task is to discover what sets excellent writing apart from merely good writing. The corpus is a collection of popular science articles published in the New York Times, in which articles by distinguished writers[1] are marked as VERY GOOD and the rest as GOOD. Where existing work on text quality related tasks (e.g. essay grading) aim at identifying text diverging from the standard in a negative way, for instance by containing spelling or grammar errors, this task is about identifying text that is different in a positive way.

One new feature group introduced by Louis and Nenkova [105] tracks the use of words belonging to coherent visual topics. Science journalists need to convey understanding of complex topics without resorting to technical terms or dry language, which would lose reader interest, and a key tool for doing this is the use of visual language. Precisely the same ability is needed in providing a good answer to a complicated question. Indeed, 7.4% of "best answers" nominations cites cognitive reasons, such as a clear explanation that makes immediate sense or which offers a novel perspective on the question [86].

1 A distinguished writer is a person whose work was featured in a yearly anthology of excellent science journalism within a ten year period.

5

QUERY-FOCUSED SUMMARIZATION

In this section we consider ideas from summarization that might help us further our goal of getting better answers.

5.1 QA AND SUMMARIZATION

When we go beyond factoids and start accepting complex questions, text summarization and question answering begin to look very much alike. It has even been suggested that *query-focused multi-document summarization* and non-factoid QA will eventually converge to one field and in the meantime would benefit from sharing experiences and approaches [98]. For example, QA has developed methods for determining relevancy with respect to a user's request, which is needed for query-focused summarization. From the other side summarization contributes the long time experience in evaluating free-text summaries where no single gold standard exists, a situation also faced by QA systems tackling complex questions.

5.2 FROM GENERIC TO QUERY-FOCUSED SUMMARIZATION

Query-focused summarization is done with respect to user input and thus differs from generic summarization in that now two factors go into deciding whether to include a piece of content or not:

1. the importance for the summary
2. the relevancy to the user's query.

In the simplest case relevancy could be estimated by cosine similarity between content and query [131].

Query-focused summarization has been a recurrent task at the evaluation conference DUC (and later TAC) since its debut in DUC 2003 [103]. The query here is not a simple, single-sentence question – the most common scenario in QA – but a "user narrative" of varying complexity. Below is an example from DUC 2007.

> *Israel / Mossad "The Cyprus Affair"* Two alleged Israeli Mossad agents were arrested in Cyprus. Deter-

> mine why they were arrested, who they were, how the situation was resolved and what repercussions there were. (D0710C)

5.3 COHERENCE IN EXTRACTIVE SUMMARIES

An extractive summarizer works by picking salient content units (e.g. sentences) from source documents and placing them in a new context in the summary. The fact that the context changes is worth noting. If the source texts were constructed simply by amassing unconnected facts, compiling a summary would be much easier, since we would not have to worry about text coherence. However, a text written by a human is, as a rule, purposely constructed to deliver a coherent message, which the strategy of choosing sentences by importance is unlikely to preserve.

As a result automatic summaries are not very readable. At DUC 2007, the system summaries were evaluated on five measures pertaining to readability. Besides grammaticality and non-redundancy, which are not considered here, they were *focus*: the ability to stay on topic; *referential clarity*, meaning pronouns and noun phrases should resolve easily; and *structure and coherence*, which is the requirement of progression in the text and local coherence between sentences. About 47% rated "barely acceptable" or lower in focus, 48% in referential clarity, and 73% in structure and coherence[1]. And things have yet to improve. At TAC 2010, automatically generated summaries scored 2.8 out of 5.0 on readability, while the average for manual summaries was 4.9[2].

The lack of coherence is partly due to the fact that sentence scoring models in summarization often have no notion of sentence order – what they deliver, in essence, is a bag of sentences. Finding an ordering of the sentences for presentation therefore needs to be done post hoc[3]. Strategies for information ordering in multi-document summarization include ordering by event time [10] and finding the ordering that optimizes local coherence relationships between sentences, either via entity grids [9]

1 http://www-nlpir.nist.gov/projects/duc/data/2007_data.html

2 http://www.nist.gov/tac/publications/2010/presentations/TAC2010_Summ_Overview.pdf

3 The linear order of sentences in a summary is important for coherence. For instance, it might be confusing to describe chronological events out of order.

or models tracking whether entity mentions are old or new in the discourse [53].

Another source of disfluency in summaries are anaphoric expressions: a sentence selected for inclusion often contains expressions referring back to text not included in the summary. To fix this, several types of such expressions must be handled:

- dangling pronouns
- definite references
- discourse markers

How much attention systems give to these phenomena varies a lot. For instance, in the case of discourse markers, one state-of-the-art system (which we will say more about in Appendix B) simply deletes sentence-initial DMs whenever they are encountered[4]. This somewhat masks the problem, since it avoids explicitly introducing misleading relations between sentences, but in the larger picture it contributes little towards a solution.

To produce more coherent summaries, it might be necessary to reconsider the way we assemble them. Say we have determined, by some means, that a sentence is relevant and needs to go in the summary. At this point, perhaps we should stop and ask: *What else do we need to include for that sentence to make sense?* This of course complicates matters. For example, a pronoun might not resolve and thus the sentence where it was introduced must be marked as a dependent. But since dependencies are transitive (for sentences a, b, and c: if a depends on b, and b depends on c, then a depends on c), the final tree of dependencies might be extensive. Also, the relevancy score for the whole tree will be different from the score of the sentence, which in turn will affect our ability to include key sentences. Therefore, it becomes necessary to consider compromises: Should pronouns sometimes be replaced by their reference? Can a DM be left out and if not, is it possible to include a summarized version of its arguments? Can the parts of the sentence containing expensive references be "compressed" away without also losing the bits that made it worth including?

5.4 DISCOURSE-BASED APPROACHES

A discourse-based summarizer works from a representation of the informational and argumentative structure of a document,

4 The behavior can be observed by examining the publicly available source code of Berg-Kirkpatrick et al. [13].

such as those obtained from a discourse parser, and uses the relations of the structure to discover the most salient content. In Rhetorical Structure Theory (RST) [108], for example, discourse relations are in many cases asymmetrical: central spans of text are called *nuclei*, while more peripheral spans are referred to as *satellites*. An example of this is the EVIDENCE relation, which holds between a nucleus and a satellite, such that the satellite provides evidence for the nucleus. More precisely, the satellite presents facts that the reader is ready to accept or already knows to be true, with the intent of increasing the reader's belief in the nucleus. Ex. (5.26) shows an example of the relation; spans are marked by brackets and the superscripted letter indicates the argument type.

(5.26) [The truth is that the pressure to smoke in junior high is greater than it will be any other time of one's life]^{N}: [we know that 3,000 teens start smoking each day]^{S} [110]

Roughly speaking, an RST-based summarizer works by selecting nuclei and leaving out satellites[5]. In theory, using discourse structure result in more readable summaries because the nucleus is supposed to be more independent in comparison with the satellite, which often cannot be understood without reference to the nucleus. Furthermore, the nucleus is thought to be more important than the satellite for the purposes of the writer. [166].

Summarization by discourse structure is thus different from the approach discussed in Section 5.3, where sentences are scored independently of how they are embedded in the deep structure of the document[6]. According to a recent evaluation by Louis et al. [106], however, it is not inherently better. They investigate correlations between discourse structure and content selection power in summarization. The method is to extract summaries using either the discourse structure (as described in this section) or the semantic sense of discourse relations, and compare

5 In reality more machinery is involved, since the basic units joined by discourse relations combine to larger units also participating in discourse relations. The tree obtained from recursively combining units is a complete cover of the document. Marcu [110] introduce the concept of a *promotion set* to describe the salient spans of any internal node in the tree.

6 Surface structure, on the other hand, often affects scoring. For example, it is not uncommon to boost the score of sentences appearing in the first paragraph of a document.

them with human summaries. Discourse information is annotated by hand, coming from the RST corpus [27], Graph Bank [182], and PDTB [139], which means performance represents an upper bound. Despite this, results are not encouraging. They find semantic sense is not indicative of important content, although it may help to filter out certain kinds of unwanted content. Discourse structure, on the other hand, is highly predictive of importance, but does not beat a baseline using simple lexical information. So even assuming gold discourse parses, there is no improvement over a "cheap", lexical baseline. Their conclusion is worth quoting in part:

> Therefore, we suspect that for building summarization systems, most benefits from discourse can be obtained with regard to text quality compared to the task of content selection. [106]

6

GOOD ANSWER, LONG ANSWER

We consider the problem of learning to rank answers at StackOverflow from user feedback. We propose a sampling procedure to correct a systemic bias in voting that favors early submitted answers, admitting only pairs of answers where the most highly ranked was submitted latest. We use features reported to be discriminative on similar data, along with features that describe discourse properties. The model is validated on the task of essay grading, but in the end we report a negative result: Using text length alone leads to better rankings for StackOverflow. An experiment where we asked human subjects to rate answers, controlling for length, suggests that it will be hard to do better on this data.

6.1 INTRODUCTION

The success of StackOverflow and similar community question answering sites depends heavily on user feedback that promotes high quality answers making them easy to find, and relegates answers of poor quality to the bottom of the page. User feedback is an interesting resource, which has been used by researchers to find experts [163, 187], to identify the answer selected by the asker as the best [15, 23], and to rank answers with respect to perceived quality.

Learning to predict user responses to a given item is interesting for a lot of reasons. A model that predicts user responses implicitly gives us a model to rank answer candidates outside of community-based question answering sites. Such a model can be applied to rerank the output of question-answering systems, but also to other problems of rating texts, say automatically scoring essays or political arguments.

In order to learn a model that is generally applicable, we constrain ourselves to using features that are intrinsic to the question and the answer. We do not take user information or the network structure of StackOverflow into account. StackOverflow was launched in 2008 and has attracted more than 10,000,000 answers.

The user feedback at StackOverflow takes the form of voting. There is one major caveat, however: the voting process is not fair. Below we demonstrate an early answer bias and discuss a sampling strategy to avoid it. We also point out a strong dependence between how early an answer arrives and how long it is. Section 6.3 presents our machine learning experiments validating our model on essay scoring data, and our negative result on the StackOverflow data. Section 6.4 presents our experiment with human annotators, suggesting that the problem of answer ranking in StackOverflow is unlearnable beyond what can be said from answer length alone. In the absence of world knowledge, that is.

6.1.1 *Early answer bias*

There are two reasons why the StackOverflow voting process is unfair. First, the voting activity begins before all of the answers are in. Second, visitors to the question page will be presented with the currently most highly ranked answer at the top. Fewer voters are therefore likely to be exposed to the low-ranking answers, even after the answer set is complete. In addition, a recent study in psychology shows that in environments where quick decisions are needed, people display an overwhelming preference for the first available choice [28].

Figure 3 illustrates the early answer bias. All answers that arrive after the initial answer (time rank > 1) have a below-expected chance of being ranked highest. Although the offset between the theoretical and observed lines remains almost constant across all time ranks, it actually becomes progressively more difficult for answers to rise to the top, because the relative difference between the two values increases. For instance, the observed chance for an answer with time rank = 9 is 3.2%, a drop of 59.7% compared with the theoretical value of 8.0%, whereas the drop for time rank = 3 is only 33.6%.

6.1.2 *Length and time rank*

Several studies have documented a link between answer length and reported quality [1, 18]. Harper et al. [68] show that on Yahoo! Answers, length and number of hyperlinks (which probably also correlates with length) alone explain a substantial part of the variance in their data. Also working on data from Yahoo! Answers, Blooma et al. [18] find length to be among the

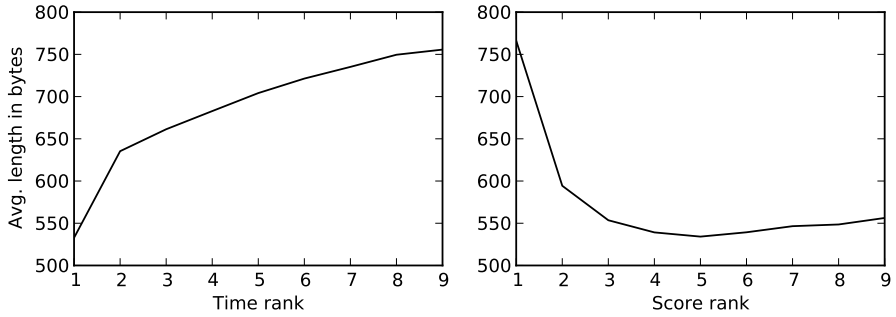


Figure 2: *Left:* Average answer length as a function of time rank. *Right:* Average answer length as a function of score rank.

top predictors for answer quality. That longer answers are also perceived of as better at StackOverflow, is clear from Figure 2, where the right panel shows the average length steadily decreasing for lower scored answers. Note that around score rank = 5 there is a slight bump; we will see the reason for this in a moment. The left panel shows another trend: The more answers already in a thread, the longer the next answer will be. Given this information one might reasonably expect that answers that arrive late (and therefore have long average length) would be frequent among the top scoring answers, but as mentioned above the opposite is actually true.

The graph in Figure 3, which shows the average length of top-ranked answers by time rank, suggests a different story. It reveals an almost linear relationship between answer length and arrival time. If the answer arrives early, it can be short and still be voted as the top answer, but if it arrives late it takes a much longer answer to secure the top position.

Anderson et al. [6] propose the pyramid as a mental image of the answering process on StackOverflow. Starting from the top, a high reputation user from a small group of very frequent visitors to the site delivers the first answer, usually within a few minutes after the question is posted. After that, medium reputation users join in and, as time goes by, the user group broadens. However, the fact that early answers attract more votes and therefore lead to higher reputation scores questions this model. Is it really the case that the expert users answer first, or do the users that deliver fast answers simply become high reputation users for just that reason?

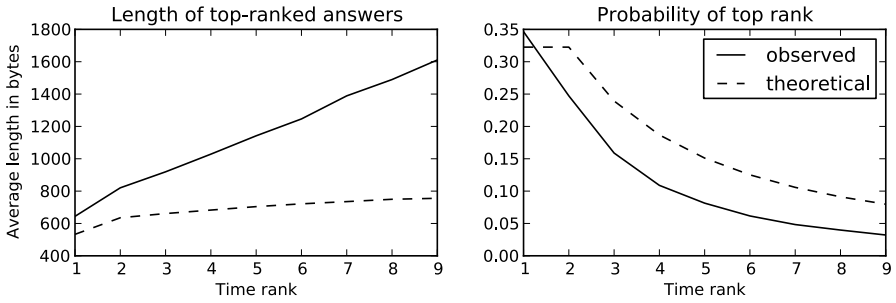


Figure 3: *Left:* Length of top-ranked answers. For comparison the dashed line shows the average length irrespective of score rank. *Right:* The solid line shows the percentage of answers that reach top score rank, plotted against time rank. The theoretical percentage is calculated as $1/n_i$ where n_i is the average answer count for threads with at least i answers.

6.2 EXPERIMENTS

6.2.1 *StackOverflow*

We use the pairwise approach to ranking, in which ranking is transformed into binary classification [71]. In pairwise ranking, two items a and b are compared, and the feature vector for the comparison is the difference between the feature vectors of the items. Let $\Phi(\cdot)$ denote the feature function. Then $\Phi_{a \prec b} = \Phi(a) - \Phi(b)$. The interpretation of the model is that we learn the relation "ranks ahead of".

We take two steps to ensure that the contrast between the answers is real, and not the result of any early answer bias:

- a must be published after b, ensuring that a voter at least had the opportunity to read b before voting on a
- a must have at least twice the number of votes of b

We refer to this as CONSERVATIVE sampling. Although applying these criteria discards a large portion of the training data, it still leaves a set of $N = 95,769$ contrastive pairs.

To control for answer length we perform an additional experiment, where each pair of answers are adjusted to be of the same length. We leave out the middle part of the longer answer, which is preferable to truncating, because it avoids systematically removing text elements that usually belong near the end of an answer, such as a conclusion or wishing the answerer good luck.

The learner is L_2 regularized logistic regression, with the strength of the regularization determined by cross-validation.

6.2.2 *Essay scoring*

To validate our feature model we present results on a dataset of 12,978 graded student essays, which was released on Kaggle, as part of a machine learning contest[1].

The essays are divided into eight sets of 1,500-1,800 each, with the exception of the last set, which, at 723, is somewhat smaller. Two types of essays are represented by four sets each: the persuasive essay, in which the student must present his opinion on a given topic with the aim of convincing the reader, and the source dependent response where the student reacts to a given passage of text. All essays are in English and written by 7th to 10th grade students.

As the score range varies by essay set, scores are normalized to fit in the 1-10 interval to make results easier to interpret. We report mean absolute error on the normalized scale and explained variance, r^2. To predict the scores we fit a linear regression model with L_2 regularization, determining the strength of the regularization by five-fold cross-validation. Each essay set is randomly split into a training (75%) and a test part (25%). We perform feature selection via χ^2 and include the 50,000 highest scoring features.

In this dataset length also has a large impact on the outcome. The correlation between essay length in bytes and grade is $r = .71$, averaged over essay sets, which have different average lengths.

6.2.3 *Feature model*

Our feature model comprises a range of shallow text features reported to be discriminative on similar data, along with features of discourse properties. The citations refer to studies using the same features for similar tasks. In addition, we also evaluated an augmented model with disambiguated discourse connectives and measures for sentence complexity, but results did not differ significantly from those obtained with the simpler model presented here.

1 http://kaggle.com/c/asap-aes

Features	r^2	Mean absolute error
Length	.519	1.158
+Patterns+Categories	.585	1.036
Full	.620	.987

Table 7: Results for essays. r^2 is explained variance (higher is better). Mean absolute error is measured on a 1-10 scale (lower is better).

LENGTH Number of tokens and bytes [77], number of content words present in answer, but not in the question, as well as non-stopword overlap between question and answer [187]

FORMATTING Rate of HTML formatting tokens [77], and rate of links [68]

STYLE Average sentence length, average token length, rates of punctuation tokens, and Flesch Kincaid reading level.

PATTERNS Partly lexicalized patterns with discourse words [61]

CATEGORIES Pronoun use [169], words indicating various degrees of trust [159], and positive and negative emotion words [4]

6.3 RESULTS

6.3.1 *Essay scoring*

Table 7 compares three feature models. Using only the Length feature group we get $r^2 = .519$, which means that length alone accounts for more than half of the variance. Still, including Patterns and Categories results in a significant improvement, and adding more feature groups leads to further improvements.

6.3.2 *StackOverflow*

Our feature model led to significant improvements for essay scoring, but on the StackOverflow data this is not the case. Instead we see a slight although not significant fall in F_1 score

Features	Orig. length		Same length	
	F_1	Acc.	F_1	Acc.
Length	.77	.76	.38	.51
+patterns+categories	.76	.76	.56	.56
Full	.76	.77	.58	.58

Table 8: Ranking of StackOverflow answers. Accuracy and F_1 scores are for the binary discrimination task (does a rank ahead of b?).

when using our full model. In the length-controlled setting performance is almost down to random.

6.4 human evaluation

Our experiment in Section 6.3.2 suggests that length alone predicts perceived answer quality better than any of the features considered in our full model. To assess how difficult the task of predicting perceived answer quality is, we asked human annotators—faculty staff members and grad students—to make judgements for a subset of our training set of $N = 100$ answer pairs. Our annotators were asked not to check the correctness of the answer, but instead indicate which answer they found to be most convincing. For this reason the question was not included in the text we presented to the annotators. Finally, we also presented annotators with snippets of answers to control for length differences.

The agreement scores for this task are shown in Table 9. There is surprisingly little consensus given that the answer pairs are selected—using Conservative sampling—so that the difference in quality between the answers is large (measured by the number of votes). The average pairwise agreement is .61, with chance agreement .50.

Note that in general the agreement between pairs of annotators is larger than the agreement between the community votes and annotators. One likely explanation for this is that the answers shown to the annotators were controlled for length.

	A_1	A_2	A_3	A_4	A_5	Gold
A_1	-	.66	.64	.64	.61	.58
A_2	-	-	.70	.60	.61	.54
A_3	-	-	-	.62	.61	.51
A_4	-	-	-	-	.63	.60
A_5	-	-	-	-	-	.57

Table 9: Observed agreement for pairs of annotators A_i and the "gold" voting scores from StackOverflow. Agreement by chance is 0.50.

6.5 CONCLUSION

We built a model for ranking answers by quality on StackOverflow. We pointed out an early-answer bias that gives unfair advantage to the first answers in a thread and presented a sampling procedure that corrects it. However, our model of both proven features and new discourse-related features was unable to beat a simple length baseline, although it did significantly improve the results on a related essay grading task. This lead us to suggest that perhaps the task is simply not learnable without world knowledge. Finally, a human evaluation showed a remarkably low agreement on the same task, confirming the picture.

Part II

ARTICLES

7

CROSS-DOMAIN ANSWER RANKING USING IMPORTANCE SAMPLING

ABSTRACT

We consider the problem of learning how to rank answers across domains in community question answering using stylistic features. Our main contribution is an importance sampling technique for selecting training data per answer thread. Our approach is evaluated across 30 community sites and shown to be significantly better than random sampling. We show that the most useful features in our model relate to answer length and overlap with question.

7.1 INTRODUCTION

Community Q&A (cQA) sites are rich sources of knowledge, offering information often not available elsewhere. While questions often attract the attention of experts, anyone can chip in, and as a result answer quality varies a lot [54]. cQA sites deal with this problem by engaging the users. If people like an answer or find it useful, they vote it up, and if it is wrong, unhelpful or spammy, it gets a down vote and is sometimes removed altogether. To a large degree the success of cQA can be attributed to this powerful content filtering mechanism. The voting induces a ranking of the answers, and that is the ranking we wish to reproduce in this paper.

We are interested in learning a ranking model based on textual or stylistic features only, extracted from the question and the answer candidate, because willfully ignoring information about user behavior and other social knowledge available in cQA sites makes our model applicable in a wider range of circumstances. Outside the world of cQA, automatic answer ranking might, for instance, be used to prioritize lists of answers found in FAQs or embedded in running text. In other words, we are interested in learning *a reranking model that is generally applicable to question answering systems*.

Part of what makes one answer preferable to another is how effective it is in communicating its advice. There may be plenty

of answers that in some technical sense are correct and yet are not especially helpful. For instance, if the kind of advice we are looking for involves a procedure, an answer structured as "First ... Then ... Finally" would probably be of greater use to us than an answer with no discernible temporal structure. Our features capture aspects of the discourse surface structure of the answer. If the model is supposed to be generally applicable to question answering it also needs to exhibit *robust performance across domains*. Learning that mentions of specific Python modules correlate with answer quality in StackOverflow does not help us answer questions in the cooking domain. We need to limit ourselves to features that transfer across domains. We further hypothesize a link between question type and answer structure (e.g. good answers to how-to questions look different from good answers to questions that ask for definitions), and test this experimentally by choosing training data for our ranker according to question similarity.

Our contribution is thus two-fold. We evaluate various stylistic feature groups on a novel problem, namely cross-domain community answer ranking, and introduce an importance sampling strategy that leads to significantly better results.

SETUP Given a question and a list of answers the task is to predict a ranking of the answers matching the ranking induced by community voting. We approach this as a pairwise ranking problem, transforming the problem into a series of classification decisions of the form: does answer a rank ahead of b? We wish to train a model that maintains good performance across domains, and our evaluation reflects this goal. We use a leave-one-out procedure where one by one each domain is used to evaluate the performance of a ranking model trained on the rest of the domains. Testing is thus always out-of-domain, and the setup promotes learning a generic model because the training set is composed of a variety of domains.

The rest of the paper is organized as follows. In the next section we introduce the cQA corpus. Section 7.3 describes several classes of motivated, domain-independent features. Our experiments with ranking and domain adaptation by similarity are described in Section 7.5, and the results are discussed in Section 7.6. Before the conclusion we review related work in Section 7.8.

7.2 THE STACKQA CORPUS

We collected a corpus, the StackQA corpus, consisting of questions paired with two or more answers from 30 individual cQA sites on different topics[1]. All sites are a part of the Stack Exchange network, sharing both the technical platform and a few very simple guidelines for how to ask a question. In the FAQ section of all sites, under the heading of "What kind of questions should I not ask here?", an identical message appears: "You should only ask *practical, answerable questions based on actual problems that you face*. Chatty, open-ended questions diminish the usefulness of our site and push other questions off the front page." It is, in other words, not a discussion club, and if a dubious question or answer enters the system, the community has various moderation tools at disposal. As a consequence, spam is almost non-existent on the sites.

7.3 FEATURE SETS

Below we describe our six groups of features. Previous studies have shown that most of these features are correlated with answer quality, see [77, 187, 68, 159, 4].

DISCOURSE We use the discourse marker disambiguation classifier of Pitler and Nenkova [136] to identify discourse uses. We have features which count the number of times each discourse marker appears.

LENGTH This group has four features that measure the length of the answer in tokens and sentences as well as the difference between the length of the question and the length of the answer. An additional two features track the vocabulary overlap between question and answer in number of lexical items, one including stop-words and one excluding these.

LEXICAL DIVERSITY An often used measure of lexical diversity is the type-token ratio, calculated as the vocabulary size divided by the number of tokens. We use a variation, the lemma-token ratio, which works on the non-inflected forms of the words.

1 We use the August 2012 dump from http://www.clearbits.net/torrents/2076-aug-2012

LEVEL AND STYLE For most readers understanding answers with long compound sentences and difficult words is a demanding task. We track difficulty of reading using the Flesch-Kincaid reading level measure and the closely related average sentence length and average token length. Three additional stylistic features capture the rate of inter-sentence punctuation, exclamation marks, and question marks. Finally, a feature gives the number of HTML formatting tokens.

PRONOUNS Scientific text almost never uses the pronoun "I", but other genres have different conventions. In cQA, where one person gives advice to another, "I" and "you" might feel quite natural. We capture personal pronoun use in six features, one for every combination of person and number (e.g. first person, singular).

WORD CATEGORIES These features build on groups of functionally related words. Examples of categories are transition words (213), which is a non-disambiguated superset of the discourse markers, phrases that introduce examples (49), comparisons (66), and contrast (6). Numbers in parenthesis indicates how many words there are in each category. For each category we count the number of token occurrences and the number of types.[2]

7.4 IMPORTANCE SAMPLING

The cQA sites contain abundant training data, but the sites are diverse and heteregoneous. We hypothesize that training our models on similar threads from different domains will improve our models considerably. We measure similarity with respect to direct questions, disregarding any explanatory text. One complication is that the question text may have more than one sentence with a question mark after it—in fact, each thread contains 2.2 sentences ending with question marks, on average. To assess the similarity between two question threads Q and Q', we take the maximum similarity between any of their question sentences:

$$\mathrm{sim}(Q, Q') = \max_{q \in Q, q' \in Q'} \mathrm{sim}(q, q')$$

2 The word lists are distributed as a part of the LightSIDE essay assessment software package found at http://lightsidelabs.com/

The similarity function used is a standard information retrieval TF*IDF-weighted bag-of-words model. Table 10 shows an example of the similar questions found by this method.

Since importance sampling requires a separately trained classifier for each question thread, we evaluate on a small set of 500 question threads per domain.

7.5 EXPERIMENTS

For each site we sample up to 5000 question threads that contain between 2 and 8 answers. When more than one answer have the same number of votes, making it impossible to rank the answers unambigously, one of the tied answers is kept at random. The number of threads used for training is varied from 50 to 5000 to obtain learning curves. We compare importance sampling against random sampling. Because this procedure is random, we repeat it three times and report an average performance figure.

The baseline for evaluating our feature model is a TF*IDF weighted bag-of-words model with each answer normalized to unit length.

We rank the answers by applying the pairwise transformation [71] and learn a classifier for the binary relation $\prec$ ("ranks ahead of"). Training data consists of comparisons between pairs of answers in the same thread.

We report F_1 score for the binary discrimination task and Kendall's τ for the ranking. In Kendall's τ 1.0 means perfect fidelity to the reference ordering, -1.0 is a perfect ordering in reverse, and .0 corresponds to a random ordering.

7.6 RESULTS

Table 12 shows that importance sampling leads to significantly better results.

The ablation results in Table 11 show that the largest negative impact comes from removing the length-related features. Leaving them out, the performance drops to .136 (from .210) in the ranking fidelity measure.

Question
How do you clean a cast iron skillet? (Cooking)
How do you clear a custom destination? (Gaming)
How do you restore a particular table in MySQL? (DB)
How Do You Determine Your Hourly Rate? (Programmers)
Do you know how to do that? (Unix)
How do I do this? (Gaming)
How do you select the Fourth kill streak? (Gaming)
How do you deal with unusually long labels? (Ux)
How do I delete a tumblr blog? (Web apps)
How do you use your iPod shuffle or nano? (Apple)
So, how do you explain spinning tops to a nine year old? (Physics)

Table 10: The 10 questions most similar to the question in bold, not counting questions from the same domain.

	F1	τ
Full model	.593	.210
- lexical diversity	.592	.209
- discourse	.605	.235
- length	.555	.136
- level and style	.592	.211
- pronouns	.593	.210
- word categories	.600	.226

Table 11: Feature ablation study on the importance weighted system (System+Sim). The results are for a training set of 500 threads.

Thread count	Kendall's τ			F1		
	Baseline	System	System+Sim	Baseline	System	System+Sim
50	.070	.075	.099	.355	.522	.536
100	.107	.084	.129	.381	.528	.551
250	.121	.095	.166	.518	.533	.571
500	.135	.124	.199	.529	.549	.588
1000	.146	.158	.229	.557	.566	.603
5000	.161	.215	**.253**	.578	.595	**.615**

Table 12: Ranking performance. Baseline is a bag-of-words model, and System uses the full feature set described in the paper. System+Sim uses the same feature model as System but with importance sampling. Results are an average over domains, and all differences between System+Sim and System are significant at $p < .01$ using the Wilconox ranksum test.

7.7 DISCUSSION

The fact that no feature group independently contributes to the classification performance, apart from the length related features, is interesting, but note that even with the length related features removed, the system is still significantly better than the bag-of-words baseline.

The relatively low performance raises two questions, discussed below. How much trust should we put into the user rankings, which are the gold standard in the experiments? And what is the maximum performance we can expect?

There is no guarantee that people who submit votes are experts. For this reason, Fichman [54] dismiss the "best answer" feature of cQA, adding that askers often select the best answer guided by social or emotional reasoning, rather than by facts. In a case study on StackOverflow (part of the StackExchange network), Anderson et al. [6] find that voting activity on a question is influenced by a number of factors presumably not connected to answer quality, such as the time before the first answer arrives, and the total number of answers.

With respect to the maximum attainable performance, an important consideration is that an answer is judged on other factors than how well it is written. When seeking a solution to a practical problem, the best answer is the one that solves it, no matter how persuasive the other answers are. This holds par-

ticularly true for cQA sites that advice people only to ask questions related to actual, solvable problems. The textual model is strong mainly if we have multiple alternative answers, which are indistinguishable with respect to facts, but differ in how their explanations are structured.

7.8 RELATED WORK

Moschitti and Quarteroni [127] consider the problem of reranking answers in question-answering systems. They use kernelized SVMs, noting that the kernel function between (question, answer) pairs can be decomposed into a kernel between questions and a kernel between answers:

$$K(\langle q, a\rangle, \langle q', a'\rangle) = K(q, q') \oplus K(a, a')$$

They share the intuition behind our approach, that pairs with more similar questions should have heigher weight, but we sample data points instead of weighting them and use different similarity functions. Choi et al. [32] establish a typology of questions in social media, identifying four different varieties: information-seeking, advice-seeking, opinion-seeking, and non-information seeking. For our purposes their categories are probably too broad to be useful, and they require manual annotation.

Agichtein et al. [2] identify high quality answers in the Yahoo! Answers data set. In addition to a wide range of social features, they have three groups of textual features: punctuation and typos, syntactical and semantic complexity, and grammaticality.

Shah and Pomerantz [152] evaluate answer quality on Yahoo! Answers data. They solicit quality judgements from Amazon Mechanical Turk workers who are asked to rate answers by 13 criteria, such as readability, relevancy, politeness and brevity. The highest classification accuracy is achieved using a combination of social and text length features.

Lai and Kao [93] address the problem of matching questions with experts who are likely to be able to provide an answer. Their algorithm is tested on on data from StackOverflow.

He and Alani [70] investigate best answer prediction using StackExchange's Serverfault and cooking communities as well as a third site outside the network.

7.9 CONCLUSION

In this paper we report on experiments in cross-domain answer ranking. For this task we introduced a new corpus, a feature representation and an importance sampling strategy. While the questions and answers come from a cQA setting, models learned from this corpus should be more widely applicable.

ACKNOWLEDGEMENTS

We wish to thank the ESICT project for partly funding this work. The ESICT project is supported by the Danish Council for Strategic Research.

8

USING CROWDSOURCING TO GET REPRESENTATIONS BASED ON REGULAR EXPRESSIONS

ABSTRACT

Often the bottleneck in document classification is finding good representations that zoom in on the most important aspects of the documents. Most research uses n-gram representations, but relevant features often occur discontinuously, e.g., *not...good* in sentiment analysis. Discontinuous features can be expressed as regular expressions, but even if we limit the regular expressions that we derive from a set of documents to some fixed length, the number becomes astronomical. Some feature combination methods can be seen as digging into the space of regular expressions. In this paper we present experiments getting experts to provide regular expressions, as well as crowdsourced annotation tasks from which regular expressions can be derived. Somewhat surprisingly, it turns out that these crowdsourced feature combinations outperform automatic feature combination methods, as well as expert features, by a very large margin and reduce error by 24-41% over n-gram representations.

8.1 INTRODUCTION

Finding good representations of classification problems is often glossed over in the literature. Several authors have emphasized the need to pay more attention to finding such representations [178, 46], but in document classification most research still uses n-gram representations.

This paper considers two document classification problems where such representations seem inadequate. The problems are answer scoring [25], on data from stackoverflow.com, and multi-attribute sentiment analysis [114]. We argue that in order to adequately represent such problems we need *discontinuous* features, i.e., regular expressions.

The problem with using regular expressions as features is of course that even with a finite vocabulary we can generate infinitely many regular expressions that match our documents.

			BoW		*Exp*		*AMT*	
	n	P(1)	m	μ_x	m	μ_x	m	μ_x
StackOverflow	97,519	0.5013	30,716	0.00131	1,156	0.1380	172,691	0.00331
Taste	152,390	0.5003	38,227	0.00095	666	0.10631	114,588	0.00285
Appearance	152,331	0.5009	37,901	0.00097	650	0.14629	102,734	0.00289

Table 13: Characteristics of the feature sets collected

We suggest to use expert knowledge or crowdsourcing in the loop. In particular we present experiments where standard representations are augmented with features from a few hours of manual work, by machine learning experts or by crowdsourcing.

Somewhat surprisingly, we find that features derived from crowdsourced annotation tasks lead to the best results across the three datasets. While crowdsourcing of annotation tasks has become increasing popular in NLP, this is, to the best of our knowledge, the first attempt to crowdsource the problem of finding good representations.

8.2 EXPERIMENTS

8.2.1 *Data*

The three datasets used in our experiments come from two sources, namely stackoverflow.com and ratebeer.com. The two datasets from the beer review website (Taste and Appearance) are described in McAuley et al. McAuley et al. [114] and available for download.[1] Each input example is an unstructured review text, and the associated label is the score assigned to taste or appearance by the reviewer.

We extracted the StackOverflow dataset from a publicly available data dump,[2], and we briefly describe our sampling process here. We select pairs of answers, where one is ranked higher than the other by stackoverflow.com users. Obviously the answers submitted first have a better chance of being ranked highly, so we also require that the highest ranked answer was submitted last. From this set of answer pairs, we randomly sample 97,519 pairs.

1 http://snap.stanford.edu/data/web-RateBeer.html
2 http://www.clearbits.net/torrents/2076-aug-2012

Our experiments are classification experiments using the same learning algorithm in all experiments, namely L_1-regularized logistic regression. The only differences between our systems are in the feature sets. The four feature sets are described below: *BoW*, *HI*, *Exp* and *AMT*.

For motivating using regular expressions, consider the following sentence from a review of John Harvard's Grand Cru:

(8.27) Could have been more flavorful.

The only word carrying direct sentiment in this sentence is *flavorful*, which is positive, but the sentence is a negative evaluation of the Grand Cru's taste. The trigram *been more flavorful* seems negative at first, but in the context of negation or in a comparative, it can become positive again. However, note that this trigram may occur discontinuously, e.g., in *been less watery and more flavorful*. In order to match such occurrences, we need simple regular expressions, e.g.,:

```
been.*more.*flavorful
```

This is exactly the kind of regular expressions we asked experts to submit, and that we derived from the crowdsourced annotation tasks. Note that the sentence says nothing about the beer's appearance, so this feature is only relevant in Taste, not in Appearance.

8.2.2 *BoW and BoW+HI*

Our most simple baseline approach is a bag-of-words model of unigram features (*BoW*). We lower-case our data, but leave in stop words. We also introduce a semantically enriched unigram model *(BoW)+HI*, where in addition to representing what words occur in a text, we also represent what Harvard Inquirer[3] word classes occur in it. The Harvard Inquirer classes are used to generate features from the crowdsourced annotation tasks, so the semantically enriched unigram model is an important baseline in our experiments below.

3 http://www.wjh.harvard.edu/~inquirer/homecat.htm

8.2.3 *BoW+Exp*

In order to collect regular expressions from experts, we set up a web interface for querying held-out portions of the datasets with regular expressions that reports how occurrences of the submitted regular expressions correlate with class. We used the Python `re` syntax for regular expressions after augmenting word forms with POS and semantic classes from the Harvard Inquirer. Few of the experts made use of the POS tags, but many regular expressions included references to Harvard Inquirer classes.

Regular expressions submitted by participants were visible to other participants during the experiment, and participants were allowed to work together. Participants had 15 minutes to familiarize themselves with the syntax used in the experiments. Each query was executed in 2-30 seconds.

Seven researchers and graduate students spent a total of five effective hours querying the datasets with regular expressions. In particular, they spent three hours on the Stack Exchange dataset, and one hour on each of the two RateBeer datasets. One had to leave an hour early. So, in total, we spent 20 person hours on Stack Exchange, and seven person hours on each of the RateBeer datasets. In the five hours, we collected 1,156 regular expressions for the StackOverflow dataset, and about 650 regular expressions for each of the two RateBeer datasets. *Exp* refers to these sets of regular expressions. In our experiments below we concatenate these with the *BoW* features to form *BoW+Exp*.

8.2.4 *BoW+AMT*

For each dataset, we also had 500 held-out examples annotated by three turkers each, using Amazon Mechanical Turk,[4] obtaining a total of 1,500 HITs for each dataset. The annotators were presented with each text, a review or an answer, twice: once as running text, once word-by-word with bullets to tick off words. The annotators were instructed to tick off words or phrases that they found predictive of the text's class (sentiment or answer quality). They were not informed about the class of the text. We chose this annotation task, because it is relatively easy

4 http://www.mturk.com

	BoW	*HI*	*Exp*	*AMT*
StackOverflow	.655	.654	.683	**.739**
Taste	.798	.797	.798	**.867**
Appearance	.758	.760	.761	**.859**

Table 14: Results using all features

for annotators to mark spans of text with a particular attribute. This set-up has been used in other applications, including NER [55] and error detection [41]. The annotators were constrained to tick off at least three words, including one closed class item (closed class items were colored differently from other words). Finally, we only used annotators with a track record of providing high-quality annotations in previous tasks. It was clear from the average time spent by annotators that annotating StackOverflow was harder than annotating the Ratebeer datasets. The average time spent on a HIT from the Ratebeer datasets was 44 seconds, while for StackOverflow it was 3 minutes 8 seconds. The mean number of words ticked off was between 5.64 and 6.96 for the three datasets, with more words ticked off in StackOverflow. The maximum number of words ticked off by an annotator was 41.

We spent a total $292.5 on the annotations, including a trial round. This was supposed to match, roughly, the cost of the experts consulted for *BoW+Exp*.

The features generated from the annotations were constructed as follows: We use a sliding window of size 3 to extract trigrams over the possibly discontinuous words ticked off by the annotators. These trigrams were converted into regular expressions by placing Kleene stars between the words. This gives us a manually selected subset of skip trigrams. For each skip trigram, we add copies with one or more words replaced by one of their Harvard Inquirer classes.

8.2.5 *Feature combinations*

This subsection introduces some harder baselines for our experiments, considered in Experiment #2. The simplest possible way of combining unigram features is by considering n-gram models. An n-gram extracts features from a sliding window

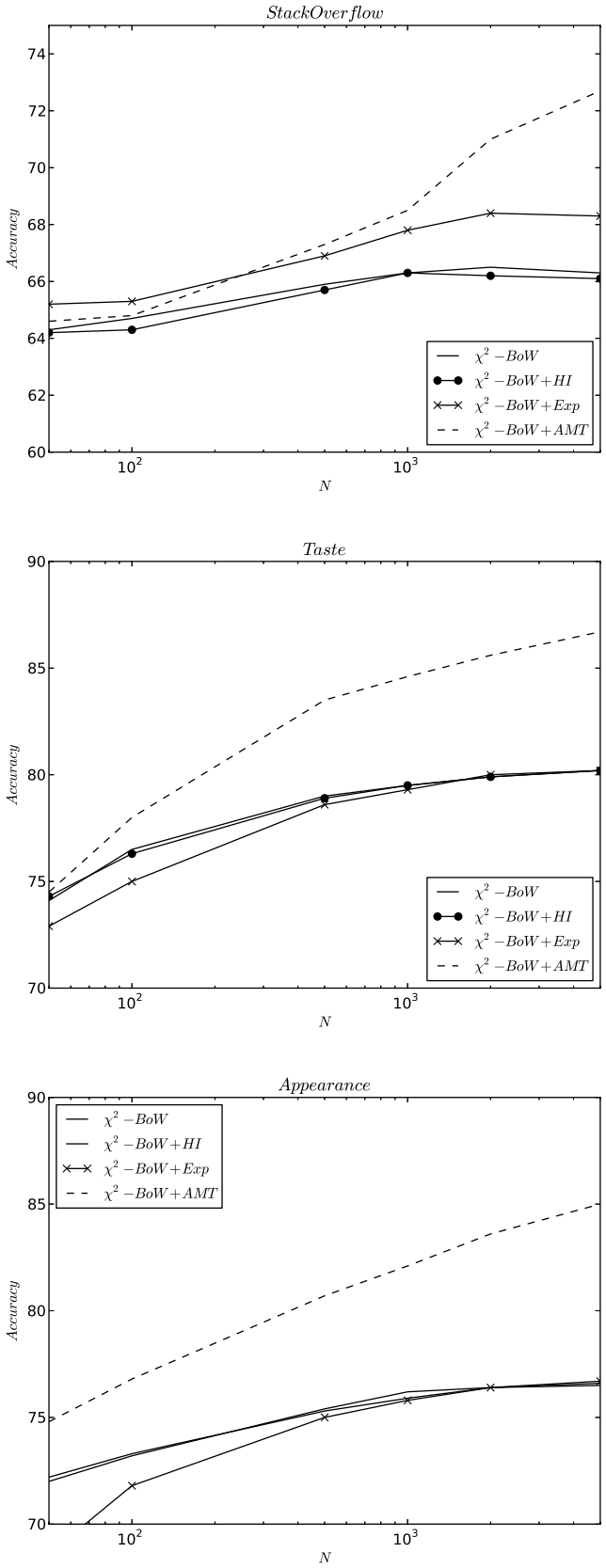


Figure 4: Results selecting N features using χ^2 (top to bottom): StackOverflow, Taste, and Appearance. The x-axis is logarithmic scale.

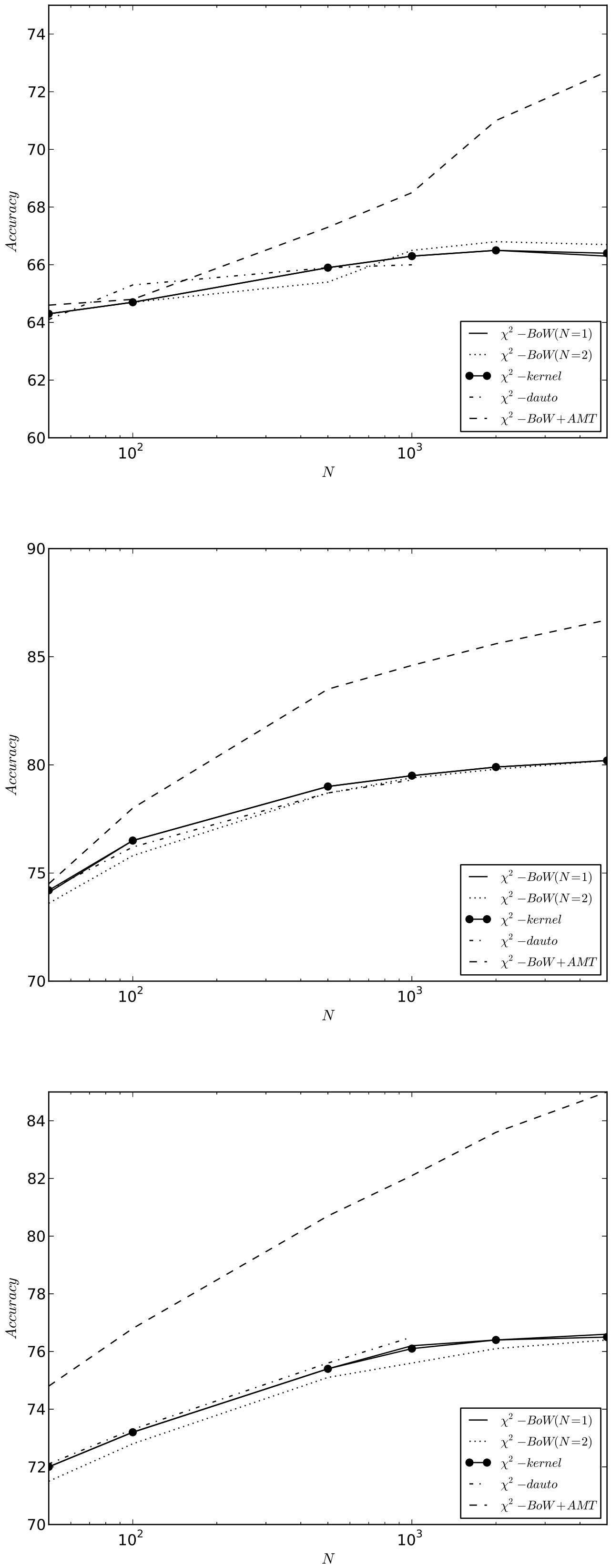


Figure 5: Results using different feature combination techniques (top to bottom): StackOverflow, Taste, and Appearance. The x-axis is logarithmic scale.

(of size n) over the text. We call this model *BoW*($N = n$). Our *BoW*($N = 1$) model takes word forms as features, and there are obviously more advanced ways of automatically combining such features.

Kernel representations We experimented with applying an approximate feature map for the additive χ^2-kernel. We used two sample steps, resulting in $4N + 1$ features. See Vedaldi and Zimmerman Vedaldi and Zisserman [176] for details.

Deep features We also ran denoising autoencoders [134], previously applied to a wide range of NLP tasks [144, 155, 30], with 2N nodes in the middle layer to obtain a deep representation of our datasets from χ^2-*BoW* input. The network was trained for 15 epochs. We set the drop-out rate to 0.0 and 0.3.

8.2.6 *Summary of feature sets*

The feature sets – *BoW*, *Exp* and *AMT* – are very different. Their characteristics are presented in Table 13. $P(1)$ is the class distribution, e.g., the prior probability of positive class. n is the number of data points, m the number of features. Finally, μ_x is the average density of data points. One observation is of course that the expert feature set *Exp* is much smaller than *BoW* and *AMT*, but note also that the expert features fire about 150 times more often on average than the BoW features. *HI* is only a small set of additional features.

8.3 RESULTS

8.3.1 *Experiment 1: BoW vs. Exp and AMT*

We present results using all features, as well as results obtained after selecting k features as ranked by a simple χ^2 test. The results using all collected features are presented in Table 14. The error reduction on StackOverflow when adding crowdsourced features to our baseline model (*BoW+AMT*), is 24.3%. On Taste, it is 34.2%. On Appearance, it is 41.0%.

Obviously, the *BoW+AMT* feature set is bigger than those of the other models. We therefore report on results using only the top-k features as ranked by a simple χ^2 test. The result curves are presented in the three plots in Figure 4. With 500 features or more, *BoW+AMT* outperforms the other models by a large margin.

8.3.2 *Experiment 2: AMT vs. more baselines*

The *BoW* baseline uses a standard representation that, while widely used, is usually thought of as a weak baseline. *BoW+HIT* did not provide a stronger baseline. In our second experiment, we show that bigram features, kernel-based decomposition and deep features do not provide much stronger baselines either. The result curves are presented in the three plots in Figure 5, and we still see that the *BoW+AMT* is significantly better than all other models with 500 features or more. Since autoencoders are consistently worse than denoising autoencoders (drop-out rate 0.3), we only plot the performance of denoising autoencoders.

8.4 RELATED WORK

Musat et al. Musat et al. [129] design a collaborative two-player game for annotation of sentiment and construction of a sentiment lexicon. One player guesses the sentiment of a text and picks a word from it that is representative of its sentiment. The other player guesses at the sentiment of the text observing only this word. If the two guesses agree, both players get a point. The idea of gamifying the problem of finding good representations goes beyond crowdsourcing, and while we did discuss several gamification strategies, this is left for future work for now. The lexicon is used in a standard representation of sentiment analysis. Boyd-Graber et al. Boyd-Graber et al. [22] crowdsource the feature weighting problem, but again within the context of standard representations. The work most related to ours is probably Tamuz et al. Tamuz et al. [168], who learn a 'crowd kernel' by asking annotators to rate examples by similarity. This kernel provides an embedding of the input examples promoting combinations of features deemed important by annotators when rating examples by similarity.

8.5 CONCLUSION

We have presented a new method for deriving feature representations from crowdsourced annotation tasks and shown that with little annotation effort, this can lead to error reductions of

24%-41% on answer scoring and multi-aspect sentiment analysis problems. On the datasets considered here, we saw no significant improvements using features contributed by experts, or using kernel representations and learned deep representations.

9

LEARNING TO DISAMBIGUATE UNKNOWN DISCOURSE MARKERS

ABSTRACT

Discourse relations hold between spans of text and are often signaled by expressions such as *although*, *moreover*, *but*, and *in comparison*, referred to as discourse markers (DM). Many researchers (e.g. Knott and Dale [91] and Halliday and Hasan [67]) have assumed that discourse markers (DMs) formed a closed class with relatively few members, spanning at most a handful of syntactical categories, but recently evidence from several sources (discussed in the paper) have established that the class is considerably larger and more diverse, and may even be open-ended. This seriously challenges the assumptions of state-of-the-art DM disambiguation and sense-tagging tools [136] and full discourse parsers [100, 60], which depend on having labeled data for all DMs. With inspiration from unsupervised domain adaptation we propose to view the DM of interest as unknown at test time and perform leave-one-out evaluation where, in each round, labeled data for one DM is selected for test and the rest is used for training. We present results for the two tasks of DM disambiguation and classification.

9.1 INTRODUCTION

Discourse relations are what makes a text come together. They provide semantic structure above the clause and sentence level, which informs many NLP tasks, including summarization [110], information extraction [124], and sentiment analysis [128]. Moreover, as we move closer to the eventual goal of machine reading [138], awareness of discourse in applications becomes increasingly important.

Although some discourse relations are not explicitly signaled (beyond adjacency), DMs remain the primary source of information about discourse structure. Being able to accurately identify these building blocks of discourse is thus necessary for constructing more complex discourse representations. However, state-of-the-art performance in DM disambiguation can only

be obtained given unrealistic assumptions about the availability of labeled training data. For instance, the number of lexically frozen, "core" markers is substantially larger than what is annotated in the PDTB (See Section 9.3.1).

The lack of specific training data for individual DMs matters, because DMs form a heterogeneous class. Fortunately, there are syntactical and semantic properties common to groups of DMs, and these similarities can be exploited to construct appropriate training data for any DM, even without having access to labeled examples of it.

SETUP We use a leave-one-out cross-validation procedure, which ensures that the same DM is never used for train and test in a fold. Effectively, the DM is treated as unknown at test time. Given labeled examples of N types of DMs, we create N different splits in which one DM is test data while the remaining $N-1$ DMs are used for training. This evaluation procedure more accurately reflects conditions in the wild, because it avoids assuming DM specific training data.

The rest of the paper is organized as follows. In the next section we introduce DM disambiguation as a transfer learning problem. Section 9.3 discusses evidence that DMs are not a closed class. Then we go on to present our approaches for DM disambiguation (Section 9.4) and sense classification (Section 9.5). Section 9.6 reports results and is followed by a discussion in Section 9.7. We review related work in Section 9.8 and conclude in Section 9.9.

9.2 TRANSFER LEARNING

A standard assumption of machine learning is that training and test data are sampled from the same underlying distribution. As we show below, this assumption must be revisited in the case of DM disambiguation. From a linguistic perspective it has been observed that no two DMs are exactly alike with respect to syntax and semantic function [175]. This puts us in the territory of *transfer learning* [132]. It is useful to think of each DM as a *domain* characterized by a probability distribution P, where, in general, $P(X) \neq P'(X)$ for pairs of DMs. In domain adaptation terms we are dealing with a multi-domain learning problem [83] meaning that the source of the transfer consists of multiple domains. A primary concern relating to this is avoiding to learn

from misleading source data, a situation referred to as negative transfer [7].

The dominating approach to discourse classification, taken by Pitler and Nenkova [136], Lin et al. [100], Ghosh [60], and others is very similar to the domain-adaption-by-feature-augmentation strategy suggested by Daumé III [43], although they differ in terminology. Pitler and Nenkova [136], for instance, create interaction variables between each marker and the majority of the features. This corresponds to having a generic domain shared between all markers (features without interaction) and a specific domain with copies of the features for each marker (interaction features). Lin et al. [100] additionally condition lexical and part-of-speech features on the marker's part of speech, in effect introducing several new and shared domains that lie somewhere between the generic and the specific domains in terms of generality.

The strategies outlined above only work if we are able to map unseen instances in the test data to domains from the training data. Assuming the marker is unseen at test time, the domain-per-marker strategy fails. We return in Section 9.6 to the question of whether part-of-speech groups the DMs in ways meaningful for the task.

What would happen if we did not do any domain adaption and only had a single generic domain? Consider Figure 6, which shows correlations between the label (is the expression a DM?) and the most frequent values of the feature "part-of-speech of previous token". For some candidate DMs a preceding comma increases the likelihood of them being DMs, while for others it is the other way around. This is also the case for NN, only the association here is stronger and more polarized. So while "part-of-speech of the previous token" is quite predictive in the context of a particular DM, its value is diminished in the generic domain, and this is true of many other features.

The literature in domain adaptation identifies three kinds of shifts that can happen between the source and the target domains. Below P_s and P_t denote the probability distributions of the source and the target domain.

Covariance shift is the situation where the distribution of the input data shifts: $P_s(X) \neq P_t(X)$. For instance, one DM might appear in a sentence-initial position 90% and occupy the center position in the remaining 10%, while another might be evenly split among those two positions.

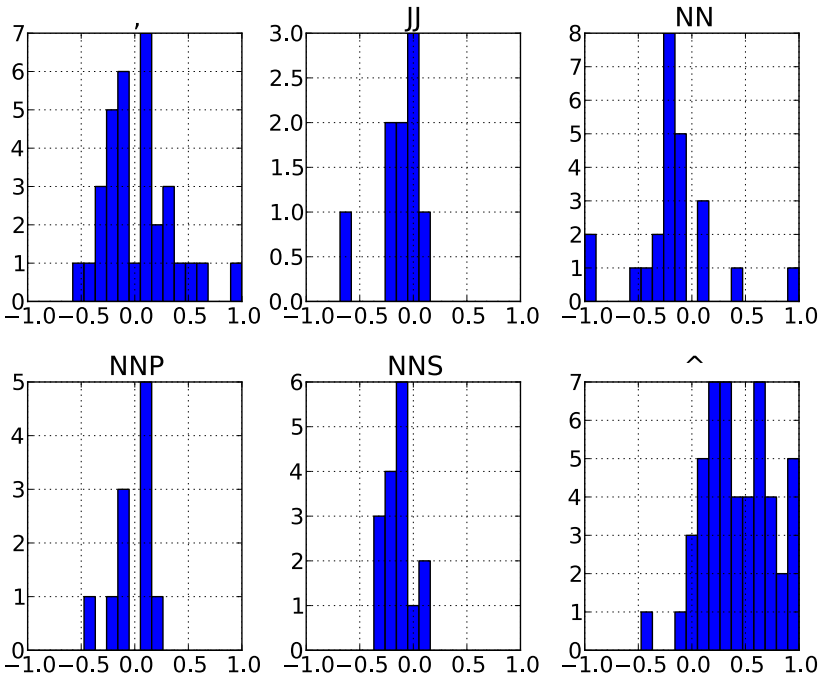


Figure 6: Histogram of correlations between part-of-speech of the token before and the label. Shown are all significant correlations (at $p < 0.1$) for the most frequent part-of-speech tags. ^ denotes the start of the sentence.

Prior probability shift happens when the distribution of the class changes: $P_s(Y) \neq P_t(Y)$. DMs vary greatly with respect to discourse ambiguity. For instance, *unless* is rarely seen in a non-discourse function (less than 10% of occurrences), while for *and* the regular uses as a conjunction far outnumbers the times it functions as a DM.

Concept shift is a change in the functional relationship between data and class: $P_s(Y|X) \neq P_t(Y|X)$. In other words the same feature value may lead to different outcomes in the source and target domains. The DMs "in addition" and "except" offer an example of this. If "in addition" is followed by the word "to", it strongly suggests that the expression is not a DM, whereas if the next word after "except" is "to", this almost surely points to the opposite conclusion.

(9.28) Janice Duclos, in addition to possessing one of the evening's more impressive vocal instruments, brings an unsuspected comedic touch to her role of Olga, everybody's favorite mom (WSJ 1163)

(9.29) The companies wouldn't disclose the length of the contract **except** to say it was a multiyear agreement (WSJ 0372)

In keeping with Daumé III et al. [44], we refer to the domain adaptation setting in which we have only unlabeled data in the target domain as *unsupervised domain adaptation*. We deal with this by clustering the DMs, thereby creating larger, more generic domains. Thus at test time the unknown DM can be mapped to a cluster for which we also have training data.

9.3 DISCOURSE MARKERS

The typical discourse marker is short, exhibits no morphological variation, and is usually spoken in a different tone from the rest of the sentence. From a syntactical point of view, discourse markers are not part of the sentence structure or only loosely connected to it [175], and they can often be removed from a sentence without making it ungrammatical:

(9.30) Everybody liked him, **since** he was always in a good mood.

(9.31) I had it, **so** I gave him a piece of my mind.

(9.32) **While** it rains today, it will be sunny tomorrow.

This is typically not the case for the same expressions when they are not used as DMs:

(9.33) Since Christmas, we have had snow every day.

(9.34) The book was so good that I read it a second time. [56]

(9.35) Texting while driving is inadvisable.

(DMs are displayed in bold-face, and non-DM expressions are underlined).

9.3.1 *DMs – an open class?*

The view that DMs form a closed class motivated the design of the PDTB, where annotators were given a fixed list of 100 lexical items and asked to disambiguate and assign senses to occurrences of these in the corpus. Automatic tools for predicting

PDTB-style discourse relations have adopted this view, including the sense-tagger and disambiguation tool from Pitler and Nenkova [136] and two full discourse parsers [100, 60]. However, results from two recent annotation efforts are beginning to change this picture.

First, evidence of a larger set of markers comes from The Biomedical Discourse Relation Bank [141], a discourse-annotated subset of the GENIA corpus [85]. Ramesh and Yu [143] find that 56% of the markers in the biomedical corpus have no counterpart in the PDTB, highlighting considerable domain differences in connective use. Some examples of the markers found exclusively in the biomedical corpus are "followed by," "due to," and "in order to".

Second, Prasad et al. [140] discuss the discovery of a number of examples from the PDTB where the discourse relation is lexicalized in an alternative way, i.e. by an expression not on the fixed list of discourse markers. (These are now annotated as AltLex: alternative lexicalizations). Ex. (9.36) and (9.37) gives examples of alternative lexicalizations.

(9.36) Cathay is taking several steps to bolster business. **One step is** to beef up its fleet. (WSJ 1432)

(9.37) The impact won't be that great, said Graeme Lidgerwood of First Boston Corp. **That is in part because of** the effect of having to average the number of shares outstanding. (WSJ 1111)

Although the group is comparatively small (624 instances), it is extremely diverse with respect to syntax. We now consider the characterization of alternative lexicalizations offered by Prasad et al. [140]. An AltLex is either,

a) syntactically admitted and lexically frozen;
b) syntactically free and lexically frozen; or
c) both syntactically free and lexically free.

The first group is perhaps least interesting. Syntactically admitted means the DM belongs to one of the accepted syntactical categories for DMs in the PDTB, and thus DMs in this groups are in a sense just "forgotten" DMs and could be added to the fixed list with no conceptual difficulty. The DMs of the second group go beyond the accepted syntactical categories but appear only in a single form and show no morphological variation. While DMs of the second group enlarge the set of DMs,

they are still in a sense fixed expressions and do not yet make it open-ended. In contrast, the last group of both syntactically and lexically free DMs are best described as partly lexicalized patterns with slots allowing for infinite combinations. For instance, the AltLex string: "A consequence of their departure could be . . . " correspond to the pattern: "<DTX> consequence (<PPX>) <VX>". Interestingly, the last group is by far the largest, making up 76.6% of the AltLex occurrences.

Additionally, new DMs emerge as a part of language change, for instance in groups of young people in Canada [167]. Variants of English are another source of differently behaved DMs, e.g. *what* in Singapore English, which marks a contrastive relationship [97]. Social media and microblogging services such as Twitter and Tumblr introduce new ways of communicating, and the space constraints of the media coupled with a low degree of formality permit constructs not otherwise seen in published writing.

(9.38) <L'année dernière à Marienbad>: Interesting (& beautiful) narration essay on memory but boy, the music is execrable **->** I almost left

(9.39) The kidlet's 1st bellydance recital vid **->** URL

Ex. (9.38) shows the iconographical DM "->" in action. "->" might be paraphrased as "consequently" because the dreadful music is what drives the author of the tweet to almost leave the performance. However, the graphical variation of the arrow in Ex. (9.39) does not appear to be used in a discourse function but rather as a focusing gesture pointing to the location of the bellydance video.

(9.40) RT @USER1: LMBO! This man filed an EMERGENCY Motion for Continuance on account of the Rangers game tonight! « Wow lmao

Gimpel et al. [63] discuss a tweet containing a similar symbol, "«", listed as Ex. (9.40). It does not, however, qualify as a DM. The initial part of the message up to "«" is a rebroadcast of a tweet from another user (RT abbreviates retweet and @USER1 is the identity of the originator), and "«" has the function of separating the added comment "Wow lmao" from the original message. As such it is pure syntax and does no work besides pointing out that the second segment comments on the first segment. In particular, it does not provide any meaning to the

relation, nor does it limit the range of possible interpretations of the second segment, both of which are thought to be defining characteristics of DMs [56].

Finally, a number of non-traditional discourse relations are established in blog entry listed as Ex. (9.41).

(9.41) In the last couple of days I've been blogging away to my heart's content ... well, as much as my partner will let me, cos we've only got one computer and you wouldn't believe the number of times we both try to use it at the same time — not to mention the power cuts — **oh yes**, they happen a lot where we live and they're a real pain, but as I say blogging away about — all sorts of things. [40]

Besides the complicated pattern of reference, the exclamation "oh yes" (in bold) is particularly interesting. Even though exclamations usually do not mark discourse relations, here it seems to be paraphrasable by "in particular", signaling an *instantiation* relation between the arguments.

Thus, as with many phenomena in language, DMs have a long-tailed distribution with a few DMs occurring very often and a long, possibly infinite, list of DMs seeing more sporadic use.

9.4 DISAMBIGUATION EXPERIMENT

Many expressions are ambiguous between a discourse function and a purely syntactical function at the type level. DM disambiguation is about determining the function unambiguously at the token level. We follow Pitler and Nenkova [136] in treating the problem as binary classification. As positive examples we use all occurrences of explicit DMs in the PDTB. Negative examples are generated by searching the PDTB for other occurrences of the same expressions. In total the dataset comprises 64,291 examples of which 18,406 are DMs.

We assign each DM to a cluster using word embeddings, which are derived on the basis of distributional information in large corpora [31]. We then apply the domain-adaption-by-feature-augmentation strategy of Daumé III [43], making cluster-specific copies of the features. Thus the feature vector of any example will have two copies of the features: one generic and one specific to the DM cluster. Algorithm 1 shows the procedure in more detail.

9.4.1 *Clustering*

We construct hierarchical clusters using Ward's mimimum-variance criterion [**?**]. The clustering can be flattened to form any number of clusters, up to the number of items in the clustering. Word embeddings represent a word as a dense vector of low dimensionality, typically between 25 and 100 dimensions, and they are constrained such that words that are distributionally similar in the corpus are also close in the vector space. We use the Eigenword[1] embeddings in the 30 dimension variant trained on trigrams from the Google ngram corpus. However, publicly available embeddings are limited in that only representations for single tokens can be obtained. In our dataset single-token DMs account for 66 out of 100 DMs, and for this experiment we chose to leave the rest out[2]. The optimal number of clusters is decided via cross-validation for each DM.

9.4.2 *Feature model*

We use several groups of syntactical features for the disambiguation task, motivated by earlier work. The constituent parse tree features and part-of-speech features were described in Pitler and Nenkova [136], and Lin et al. [100]. The dependency and position features are new to this work.

PART-OF-SPEECH The part-of-speech of the candidate, preceding token, and next token.

CONSTITUENCY These features are mainly categories of nodes in a constituent tree. They are the node completely covering the candidate expression (self category), the parent of the self category, and the left sibling and right sibling of the self category. Additionally, two indicator features for whether the right sibling of the self category contains a trace, and whether it contains a VP node.

DEPENDENCY Two features giving the part-of-speech and relation type of the head of the candidate's head. Both features heuristically assume the head is the rightmost token.

1 http://www.cis.upenn.edu/~ungar/eigenwords/

2 This is a practical consideration more than a theoretical concern: word embeddings could in principle be induced for multi-word-units like the left-out DMs.

Algorithm 1 Disambiguation algorithm

$\mathcal{D} \leftarrow$ data set
for $t \in$ DM types **do**
  $test \leftarrow \{d \in \mathcal{D} \mid d$ is example of $t\}$
  $train \leftarrow \mathcal{D} - test$
  $n \leftarrow$ find optimal cluster size using $train$
  $X_{test} \leftarrow$ BuildDomains$(test, n)$
  $X_{train} \leftarrow$ BuildDomains$(train, n)$
  train classifier on X_{train} and labels
  test classifier using X_{test}
end for

procedure BuildDomains$(instances, n)$
  Let f be the feature function
  $X \leftarrow$ empty list
  for $d \in instances$ **do**
    $L \leftarrow$ list with $n + 1$ elements
    for $i \leftarrow 1, n$ **do**
      $L_i \leftarrow \mathbf{o}$ ▷ s.t. $|\mathbf{o}| = |f(\cdot)|$
    end for
    $c \leftarrow$ map d to a cluster in an n clustering
    $L_0 \leftarrow f(d)$ ▷ generic domain
    $L_c \leftarrow f(d)$ ▷ specific domain
    flatten L and append to X
  end for
  return X
end procedure

POSITION Three binary features indicating if there is a verb left of the candidate, right of the candidate (in linear order), or above the candidate.

9.5 SENSE CLASSIFICATION EXPERIMENT

In sense classification we are given two spans of text linked by a discourse relation and are asked to determine the semantic relationship holding between the spans: the *sense* of the relation. Recall that discourse relations are either signalled explicitly by a DM or implicitly through adjacency of the arguments. In contrast to the experiments of Section 9.4, we do not use the identity of the DM, effectively treating the problem as as implicit sense classification.

In the absence of a marker, the pragmatic relationship must be inferred using lexical and semantic information from the arguments. The precise sense of the relation is probably not recoverable in this way, because disregarding the DM loses subtle (and possibly non-redundant) information [91], although other linguistic cues may make up for some of the loss [165]. For this reason we only attempt to distinguish between the senses in the top level of the PDTB sense hierarchy, comprising the four classes TEMPORAL, CONTINGENCY, COMPARISON, and EXPANSION. The same set of classes was used for sense classification of implicit relations in Pitler et al. [137] and for explicit relations in Pitler and Nenkova [136].

9.5.1 *Feature model*

Our features are based on word embeddings and different ways of composing them. First, we describe our baseline, which is a word-pair model. Marcu and Echihabi [111] were the first to use a word-pair model which includes all pairings of lexical items from the two arguments. To give an example, Ex. (9.42) show a contrastive relation (subtype *juxtaposition* in the PDTB), marked by the DM "but".

(9.42) Operating revenue rose 69% to A$8.48 billion from A$5.01 billion. **But** the net interest bill jumped 85% to A$686.7 million from A$371.1 million

Intuitively, the pair (rose, jump) suggest a parallelism between the arguments.

The main issue with word-pair models is that they do not generalize well. With the labeled training sets at our disposal, which are limited with respect to size and domain, the majority of the discriminative word pairs will never have been encountered before. Lexical resources such as Wordnet and The Harvard Inquirer provide a limited abstraction capability but only at the word level.

To overcome lexical sparseness we use word embeddings, more specifically the 80-dimensional RNN[3] representation. Word embeddings provide representations for single words, but the problem calls for representations of spans of several words, and how they interact. We use the two strategies described below.

ADDITIVE We construct a vector for each of the spans using simple additive semantics; the span vector is the mean of the word vectors. Then there are several options for composing the span vectors. We consider the difference between the vectors, and their concatenation.

SIMILARITY We adapt a method for calculating the similarity between two sentences given word-to-word similarities described in [78]. They define the similarity between two sentences $\mathbf{a}$ and $\mathbf{b}$ as

$$\mathrm{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}W\mathbf{b}^\top}{|\mathbf{a}||\mathbf{b}|}$$

Say V is the joint vocabulary of the two sentences. Then $\mathbf{a}$ and $\mathbf{b}$ are both $|V|$ length binary vectors indicating the presence/absence of terms in the vocabulary, and W is a similarity matrix in which W_{ij} gives the similarity between vocabulary items i and j. The denominator normalizes for sentence length.

In the adapted version, we stop short of actually calculating the similarity, replacing the final dot product with a component-wise multiplication, $\odot$.

$$\frac{\mathbf{a}W \odot \mathbf{b}^\top}{|\mathbf{a}||\mathbf{b}|}$$

The result is a $|V|$ vector of similarity scores for each word in the second sentence with respect to all words in the first sentence. Note that the $\mathbf{a}W$ term is a $|V|$ vector with similarities between each vocabulary item and the first sentence. The final

3 http://www.fit.vutbr.cz/~imikolov/rnnlm/

System	F1
Only generic domain	.787
Part-of-speech domains	.783
Eigenwords domains	.830

Table 15: Disambiguation results

System	F1
Word-product baseline	.400
RNN concatenation	.437
RNN similarity	.469
RNN concatenation + similarity	.489

Table 16: Sense classification results

component-wise multiplication with **b** "picks out" the vocabulary items present in the second sentence. The entries in W are populated with cosine similarities[4] between word embeddings.

9.6 RESULTS

Table 15 shows the disambiguation performance in three settings: a) without mapping to specific domains; b) mapping using the part-of-speech of the DM; and c) mapping using clusters based on Eigenwords. The Eigenwords domain mapping gives the best result, with an error reduction of 20% over the generic domain mapping and 22% error reduction with respect to part-of-speech domains. The part-of-speech mapping offers no advantage over using a single domain.

The sense classification results are listed in Table 16. Using the RNN word embeddings improves upon the baseline in all cases. The partly lexicalized similarity method (.469) gives a better result than concatenation (.437). Combining the RNN methods gives an F1 of .489, which is better than any of them individually.

4 1.0 - cosine distance

9.7 DISCUSSION

The improvements we see from inducing domains are likely the results of two effects: a *grouping* effect that comes from putting DM types with similar behavior in a single domain, and a *shielding* effect where the impact of misleading examples is mitigated when dissimilar DMs are clustered in different domains. For instance, in case of concept drift between the test DM and DMs in the training data, this will matter less if the conflicting test DMs are mapped to domains different from the test DM domain.

The cluster size most typically selected by the cross validation procedure is 40. Although many clusters at this stage only contain a single DM, some clusters with several elements remain. These are listed in Table 17. The clustering looks meaningful, grouping, for instance, {*then, therefore, thus*}. Another cluster consists of {*and, for, or*}, which are all high-frequency tokens represented by many examples in the dataset. Having them in a separate domain could contribute to the mentioned shielding effect for DMs that are dissimilar.

We use two kinds of embeddings in the experiments. Initially, we tried both Eigenwords and RNN in both experiments, and although the embeddings in all cases improved upon the baseline, Eigenwords is clearly better in disambiguation and, similarly, RNN is clearly better in sense classification. This is in line with the observation that Eigenwords work best as syntactical clusters (Dhillon et al. [45]; Lyle Ungar, p.c.), since disambiguation is mainly about syntax.

9.8 RELATED WORK

Several studies are concerned with the semantic relationships between markers. Knott [90] introduce a methodology for discovering which pairs of DMs are mutually substitutable and in what contexts, but through a labour-intensive process in which acceptability judgements are elicited from human subjects. Following up on this research, Hutchinson [75] consider the feasibility of deriving these substitutability relationships automatically by way of distributional information. They find that distributional similarity as well as a function based on variance work well for predicting the relation between markers . However, substitutability is a semantic criterion (for instance, when we replace "because" with "seeing as", does the sentence still

Cluster
although, whereas
except, unless
but, however, nor, regardless
and, for, or
besides, though
rather, so
indeed, yet
later, once
after, as, because, before, since, until, when
if, instead
then, therefore, thus
consequently, hence
furthermore, moreover
also, still
nevertheless, nonetheless
accordingly, thereafter
simultaneously, specifically

Table 17: Flat clusters. The table shows all non-singleton clusters for $N = 40$.

mean the same thing?), whereas the type of similarity of interest in the disambiguation task is grammatical.

Sporleder and Lascarides [158] consider the implicit sense classification task of Marcu and Echihabi [111], proposing a variety of features. The features fall in six groups: positional, length, lexical (including overlap), part-of-speech, temporal (verb finiteness, modality aspect, voice and negation), and cohesion features. In a more recent work, the authors consider the various features that have been proposed for this task and decide on seven groups [179]. These are polarity (sentiment), Inquirer tags (21 semantic categories), modality, repetition of same word in both spans, word pairs between, and intra-span word pairs.

9.9 CONCLUSION

In this paper we challenged a common assumption made by researchers creating discourse-related software, namely that supervised training data is available for all types of DMs. We proposed a new leave-one-out cross-validation procedure which makes the opposite assumption of no training data for any particular DM.

We argued that it is useful to think of DM disambiguation in terms of transfer learning and showed by example that covariance shift, prior probability shift, and concept shift are all likely to occur between DMs.

To support the claim that supervised data is unlikely to suffice, we reviewed several sources of additional DMs, including other text domains like scientific writing, open-ended alternative lexicalizations, and new DMs originating in social media.

In the disambiguation experiment we looked at one particular way of grouping DMs based on distributional data from word embeddings. Results improved over the baseline, and we suggested that this was due to two effects: a grouping and a shielding effect.

The sense classification experiment used word embeddings to build representations of a discourse relation's arguments. Classification results improved compared to a lexical word-pair baseline sense.

10

DISAMBIGUATING DISCOURSE CONNECTIVES WITHOUT ORACLES

ABSTRACT

Deciding whether a word serves a discourse function in context is a prerequisite for discourse processing, and the performance of this subtask bounds performance on subsequent tasks. Pitler and Nenkova [136] report 96.29% accuracy (F_1 94.19%) relying on features extracted from gold-standard parse trees. This figure is an average over several connectives, some of which are extremely hard to classify. More importantly, performance drops considerably in the absence of an oracle providing gold-standard features. We show that a very simple model using only lexical and predicted part-of-speech features actually performs slightly better than Pitler and Nenkova [136] and not significantly different from a state-of-the-art model, which combines lexical, part-of-speech, and parse features.

10.1 INTRODUCTION

Discourse relations structure text by linking segments together in functional relationships. For instance, someone might say "Saber-toothed tigers are harmless *because* they're extinct", making the second part of the sentence serve as an explanation for the first part. In the example the discourse connective *because* functions as a lexical anchor for the discourse relation. Whenever an anchor is present we say that the discourse connective is *explicit*.

Complicating the matter, phrases used as discourse connectives sometimes appear in a non-discourse function. For instance, "and" may be either a simple conjunction, as in "sugar and salt", or a discourse relation suggesting a temporal relationship between events, for instance "he struck the match and went away". The Penn Discourse Treebank (PDTB) [139] distinguish 100 types of explicit connectives—a subset of these are listed in Table 19. The type of relationship is selected from a hierachial structure where the four top-level categories are Comparison, Contingency, Temporal, and Expansion.

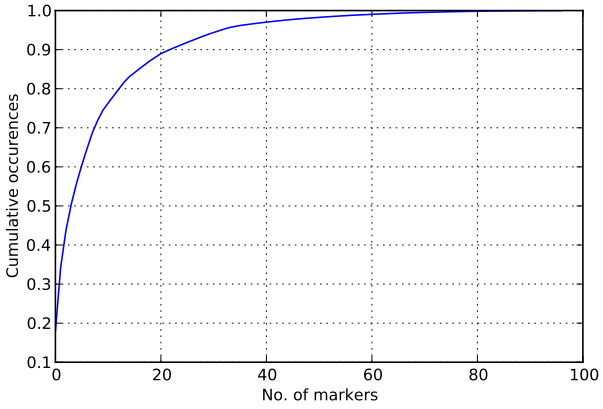


Figure 7: A picture of the problem. 10% of connectives account for roughly 75% of occurrences

Discourse relations are important for many applications and, since the PDTB was released, much effort has gone into developing tools for recreating the annotations of the resource automatically. Recently two ambitious end-to-end parsers have appeared which transform plain text to full PDTB-style annotations [100, 60]. Both systems share a pipelined architecture in which the output of one component becomes the input to the next. A crucial first step in their processing is correctly identifying explicit discourse connectives; when unsuccessful subsequent steps fail.

An accuracy in the high ninetees seems to suggest that the problem is almost solved. For the task of discourse connective disambiguation this unfortunately does not hold true, because, as we argue here, the task benefits from being seen and evaluated as a number of smaller tasks, one for each connective type. Figure 7 shows why: the distribution of connectives follows a power law such that the majority of occurrences comes from relatively few but highly frequent connective types. If we do not take into account the uneven sizes of the categories, our performance figure ends up saying very little about how well we are doing on most of the connectives, because it is being dominated by the performance on a few high-frequency items.

In this paper we look in more detail on the evaluation of the discourse connective disambiguation task, in particular how two commonly used feature models perform on individual discourse connectives. The models are Pitler and Nenkova [136] (P&N), and its extension by Lin et al. [100] (Lin). Motivated by our findings we advocate the use of macro-averaging as a neces-

sary supplement to micro-averaging. Additionally, we perform our experiments in a more realistic setting where acccess to oracle gold-standard annotations is not assumed. The observed performance drop from oracle to predicted parses leads us to propose a new model, which approximates the syntactical information of the parse trees with part-of-speech tags. Although these features are less powerful in theory, the model has comparable macro-average performance in realistic evaluation.

The rest of the paper is structured as follows. In the next section we give reasons why low-frequency connectives should not be overlooked. Section 10.3 describes our experiments, and Section 10.4 reports on the results. The discussion is in Section 10.5, followed by a review of related work in Section 10.6. Section 10.7 concludes the paper.

10.2 THE IMPORTANCE OF THE LONG TAIL

Are there any compelling reasons to pay attention to the lower-frequency connectives when high-frequency connectives overwhelmingly dominate? As noted in the caption to Figure 7, the top 10 account for above 75% of the occurrences and top 20 for above 90%. So why should we care?

It turns out that the low-frequency connectives are quite evenly distributed among texts. In the Wall Street Journal part of the Penn Treebank, 70% of articles that contain explicit markers contain at least one marker not in the top 10. Not counting very short texts (having only two or fewer explicit connectives of any type), the number rises to 87%. While low performance on less frequent connectives does not hurt a token-level macro-average much, it still means that you are likely to introduce errors in something like 70% of all WSJ articles. These errors percolate leading to erroneous text-level discourse processing.

In Webber and Joshi [180] the prime example of a discourse application is automatic text simplification. Here, ignoring the long tail of discourse connectives would be out of the question, because it is precisely those less familiar expressions — which people encounter rarely and have weaker intuitions about — that would benefit the most from a rewrite. Two other examples, also cited in Webber and Joshi [180], are automatic assessment of student essays [24], and summarization [170]. In student essays we encourage clear argumentative structure and rich vocabulary; failing to recognize that in an automatic system would not qualify as fair evaluation. And summarization is

often performed over news wire, which, as shown in the PDTB, has a high per-article incidence of connectives not in top 10. Additionally, some low-frequency connectives like "ultimately" and "in particular" are strong cues for text selection.

Another reason to suspect that low-frequency connectives are important comes from an observation about the distribution of connectives in biomedical text. Ramesh and Yu [143] report an overlap of only 44% between the connectives found in the The Biomedical Discourse Relation Bank [141], a 24 article subset of the GENIA corpus [85], and the PDTB. The intersection contains high-frequency connectives, such as "and", "however," "also," and "so". Connectives specific to the biomedical domain include "followed by," "due to," and "in order to", and the authors speculate that the unique connectives encode important domain specific knowledge.

10.3 EXPERIMENTS

Our experiments are designed to shed light on three aspects of discourse connective disambiguation: 1) error distribution wrt. connective type; uneven performance builds a strong case for averaging over connective types instead of averaging over data points; 2) performance loss in the absence of an oracle; and 3) performance of simple model based on cheaper and more reliable annotations.

We experiment with three different feature sets, all of which model syntactical aspects of the discourse connective.

The P&N and Lin feature sets are chosen to represent state-of-the-art. The high accuracy of P&N at 96.29% is frequently cited as an encouraging result, see Huang and Chen [73], Alsaif and Markert [5], Tonelli and Cabrio [171], Zhou et al. [186]. Besides discourse parsing P&N has been used for tasks as diverse as measuring text coherence [101] and improving machine translation [121]. The POS+LEX feature set is proposed as an alternative model. The baseline always predicts the majority class.

P&N This feature set derives from parse trees and replicates the features of Pitler and Nenkova [136]. Starting from the potential discourse connective, the features include the highest category in the tree subsuming only the connective called the self-category, the parent of that category, the left sibling of the self-category, and the right sibling of the self-category. A fea-

ture fires when the right sibling contains a VP, and another if there is a trace node below the right sibling. Note that the trace feature will never fire outside of the gold parse setting since state-of-the-art parsers do not predict trace nodes.

Importantly, there is a feature for the identity of the connective and interaction features between the connective and the syntactical features in effect allowing the model to fit parameters specific to each connective. Furthermore, combinations of the syntactical features are allowed, but they cannot be connective-specific.

LIN The feature set augments P&N with part-of-speech and string features for the tokens adjacent to the connective, as well as the part-of-speech of the connective itself. The part-of-speech features for the adjacent tokens interact with the part-of-speech of the connective, and the string features interact with the indicator feature for the connective. It also adds a syntax feature: the path to the root of the parse tree.

POS+LEX The simple feature set builds on part-of-speech tags and tokens. Part-of-speech tags are captured using a window of two tokens around the marker, and the lexical features are the same as Lin. Like P&N there is a feature for the identity of the connective as well as interaction features between the identity feature and other features.

In keeping with Pitler and Nenkova [136] our learner is a maximum entropy classifier trained on sections 2-22 of the WSJ using ten-fold cross-validation.

10.3.1 *Parsing Wall Street Journal*

To obtain a version of the WSJ corpus containing fully predicted parses we use the Stanford Parser[1] training a separate model for each section. To parse a specific section we train on everything but that section (e.g. for parsing section 5 the training set is section 0-4 and 6-24). Average F_1 on all sections is 85.87%. Although the very best state-of-the-art parsers[2] report F_1 of above 90%, our parsing score greatly exceeds typical performance on real-life data, which is almost always out-of-domain with re-

1 http://nlp.stanford.edu/software/lex-parser.shtml, 2012-11-12 release with the 'goodPCFG' standard settings

2 http://aclweb.org/aclwiki/index.php/?title=Parsing_(State_of_the_art)

Model	Micro		Macro	
	Oracle	Pred.	Oracle	Pred.
Baseline	72.7	72.7	53.9	53.9
P&N	93.0	90.7	85.3	80.7
Lin	95.2	92.9	86.7	83.6
POS+LEX	89.7	89.7	82.5	83.5

Table 18: Comparing F_1 score on oracle and predicted features using macro and micro averaging. A Wilcoxon signed rank test shows that the macro-averaged difference between POS+LEX and Lin10 using predicted features is not significant at $p < 0.01$.

spect to 1980s WSJ. Thus this setting still compares favourably to performance in the wild.

10.4 RESULTS

A summary of the results is found in Table 18. For a subset of frequent and less frequent connectives, Table 19 lists individual F_1 scores. In all of the feature sets we see a marked drop moving from micro-average (average over instances) to macro-average (average over connective types)—P&N, for instance, goes from 93.0% to 85.3%. This shows that the scores of less frequent connectives are somewhat lower than frequent ones. When features are derived from predicted parses performance also fall, from 93.0% to 90.7% with micro-average, and even more dramatically with macro-average, where it goes from 85.3% to 80.8%. Given that we are interested in real life performance this last figure is the most interesting.

10.5 DISCUSSION

In NLP applications we cannot assume the existence of oracles providing us with gold-standard features. Often switching to predicted features introduces greater uncertainty. If the parser often confuses two non-terminals that are important for connective disambiguation we loose predictive power. Thus, on the P&N model, the average conditional entropy per feature given the class (how surprising the feature is when we know

	Oracle		Pred.		Disc.
	Lin	P+L	Lin	P+L	
but	98.6	96.1	97.6	96.1	78.9
and	94.9	77.0	89.0	77.0	14.7
also	97.0	97.3	97.5	97.2	93.5
if	93.4	93.1	92.3	93.0	82.6
when	89.9	88.5	89.3	88.4	65.5
because	99.5	99.4	99.4	99.5	63.4
while	97.6	97.7	97.5	97.4	91.9
as	89.8	63.1	78.1	63.0	13.0
after	93.7	74.0	87.9	72.9	42.4
however	98.7	98.4	98.5	98.4	95.7
...					
ultimately	43.2	30.3	36.4	29.4	37.5
rather	84.8	83.9	80.0	83.9	8.2
in other words	97.1	94.4	91.4	94.4	89.5
as if	84.8	84.8	71.0	88.2	66.7
earlier	76.9	66.7	74.1	69.6	2.1
meantime	80.0	76.5	82.4	80.0	71.4
in particular	89.7	85.7	85.7	80.0	48.4
in contrast	100.0	100.0	100.0	100.0	50.0
thereby	95.7	95.7	100.0	95.7	100.0
...					

Table 19: F_1 score per connective. The table is sorted by the number of actual discourse connectives in the PDTB. After the break the table continues from position 50. The last column gives the percentage of discourse connectives.

the answer) increases by 8.8% when the oracle is unavailable. In contrast there is almost no difference between the conditional entropy of the POS model with oracle features and without, indicating that the errors made by the tagger are not confusing in the disambiguation task.

Predicted parse features are associated with uncertainty even when used in combination with words and part of speech. Comparing the number of times the Lin model changes an incorrect prediction of POS+LEX to a correct one and the number of times it introduces a new error by changing a correct prediction to an incorrect one, we observe that corrections almost always come with a substantial number of new errors. In fact, 58 connectives have at least as many new errors as corrections.

Predicted parse features also contribute to feature sparsity, because of the greater variability of automatic parses. On the other hand, they are more expressive than part of speech, and in the example below, where only Lin correctly identifies 'and' as a discourse connective, part of speech simply does not contain enough information.

"A whole day goes by **and** no one even knows they're alive.

10.6 RELATED WORK

Atterer and Schütze [8] present similar experiments for prepositional phrase attachment showing that approaches assuming gold-standard features suffer a great deal when they are evaluated on predicted features. Spitkovsky et al. [157] also caution against the use of gold-standard features, arguing that for unsupervised dependency parsing using induced parts of speech is superior to relying on gold-standard part-of-speech tags.

This work also relates to Manning [109] who point out that even though part-of-speech tagging accuracy is above 97% the remaining errors are not randomly distributed but in fact occur in just the cases we care most about.

10.7 CONCLUSION

Discourse connective disambiguation is an important subtask of discourse parsing. We show that when realistic evaluation is adopted — averaging over connective types and not relying on oracle features — performance drops markedly. This suggests that more work on the task is needed. Moreover, we show

that in realistic evaluation a simple feature model using part-of-speech tags and words performs just as well as a much more complex state-of-the-art model.

ACKNOWLEDGEMENTS

We wish to thank the ESICT project for partly funding this work. The ESICT project is supported by the Danish Council for Strategic Research.

11

DOWN-STREAM EFFECTS OF TREE-TO-DEPENDENCY CONVERSIONS

ABSTRACT

Dependency analysis relies on morphosyntactic evidence, as well as semantic evidence. In some cases, however, morphosyntactic evidence seems to be in conflict with semantic evidence. For this reason dependency grammar theories, annotation guidelines and tree-to-dependency conversion schemes often differ in how they analyze various syntactic constructions. Most experiments for which constituent-based treebanks such as the Penn Treebank are converted into dependency treebanks rely blindly on one of four-five widely used tree-to-dependency conversion schemes. This paper evaluates the down-stream effect of choice of conversion scheme, showing that it has dramatic impact on end results.

11.1 INTRODUCTION

Annotation guidelines used in modern dependency treebanks and tree-to-dependency conversion schemes for converting constituent-based treebanks into dependency treebanks are typically based on a specific dependency grammar theory, such as the Prague School's Functional Generative Description, Meaning-Text Theory, or Hudson's Word Grammar. In practice most parsers constrain dependency structures to be tree-like structures such that each word has a single syntactic head, limiting diversity between annotation a bit; but while many dependency treebanks taking this format agree on how to analyze many syntactic constructions, there are still many constructions these treebanks analyze differently. See Figure 8 for a standard overview of clear and more difficult cases.

The difficult cases in Figure 8 are difficult for the following reason. In the easy cases morphosyntactic and semantic evidence cohere. Verbs govern subjects morpho-syntactically and seem semantically more important. In the difficult cases, however, morpho-syntactic evidence is *in conflict* with the semantic evidence. While auxiliary verbs have the same distribution

Clear cases		*Difficult cases*	
Head	Dependent	?	?
Verb	Subject	Auxiliary	Main verb
Verb	Object	Complementizer	Verb
Noun	Attribute	Coordinator	Conjuncts
Verb	Adverbial	Preposition	Nominal
			Punctuation

Figure 8: Clear and difficult cases in dependency annotation.

as finite verbs in head position and share morpho-syntactic properties with them, and govern the infinite main verbs, main verbs seem semantically superior, expressing the main predicate. There may be distributional evidence that complementizers head verbs syntactically, but the verbs seem more important from a semantic point of view.

Tree-to-dependency conversion schemes used to convert constituent-based treebanks into dependency-based ones also take different stands on the difficult cases. In this paper we consider four different conversion schemes: the Yamada-Matsumoto conversion scheme YAMADA,[1] the CoNLL 2007 format CONLL07,[2] the conversion scheme EWT used in the English Web Treebank [135],[3] and the LTH conversion scheme [80].[4] We list the differences in Figure 9. An example of differences in analysis is presented in Figure 10.

In order to access the impact of these conversion schemes on down-stream performance, we need extrinsic rather than intrinsic evaluation. In general it is important to remember that

1 The Yamada-Matsumoto scheme can be replicated by running penn2malt.jar available at http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html. We used Malt dependency labels (see website). The Yamada-Matsumoto scheme is an elaboration of the Collins scheme [35], which is not included in our experiments.

2 The CoNLL 2007 conversion scheme can be obtained by running pennconverter.jar available at http://nlp.cs.lth.se/software/treebank_converter/ with the 'conll07' flag set.

3 The EWT conversion scheme can be replicated using the Stanford converter available at http://http://nlp.stanford.edu/software/stanford-dependencies.shtml

4 The LTH conversion scheme can be obtained by running pennconverter.jar available at http://http://nlp.cs.lth.se/software/treebank_converter/ with the 'oldLTH' flag set.

Form 1	*Form 2*	YAMADA	CONLL07	EWT	LTH
Auxiliary	Main verb	1	1	2	2
Complementizer	Verb	1	2	2	2
Coordinator	Conjuncts	2	1	2	2
Preposition	Nominal	1	1	1	2

Figure 9: Head decisions in conversions. Note: YAMADA also differ from CONLL07 wrt. proper names.

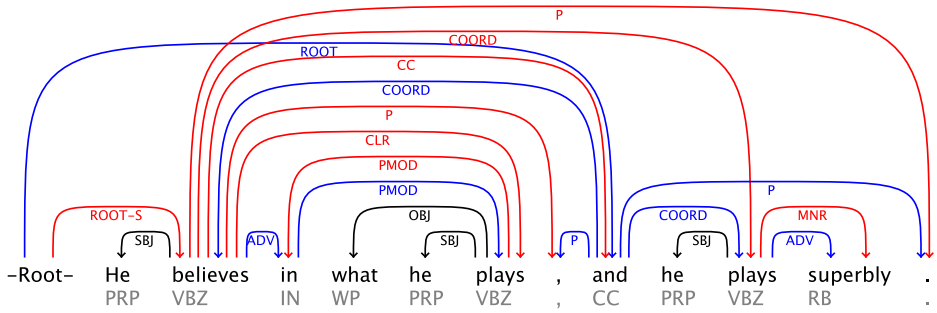


Figure 10: CoNLL 2007 (blue) and LTH (red) dependency conversions.

while researchers developing learning algorithms for part-of-speech (POS) tagging and dependency parsing seem obsessed with accuracies, POS sequences or dependency structures have no interest on their own. The accuracies reported in the literature are only interesting insofar they correlate with the usefulness of the structures predicted by our systems. Fortunately, POS sequences and dependency structures *are* useful in many applications. When we consider tree-to-dependency conversion schemes, down-stream evaluation becomes particularly important since some schemes are more fine-grained than others, leading to lower performance as measured by intrinsic evaluation metrics.

Approach in this work

In our experiments below we apply a state-of-the-art parser to five different natural language processing (NLP) tasks where syntactic features are known to be effective: negation resolution, semantic role labeling (SRL), statistical machine translation (SMT), sentence compression and perspective classifica-

tion. In all five tasks we use the four tree-to-dependency conversion schemes mentioned above and evaluate them in terms of down-stream performance. We also compare our systems to baseline systems not relying on syntactic features, when possible, and to results in the literature, when comparable results exist. Note that negation resolution and SRL are not end applications. It is not easy to generalize across five very different tasks, but the tasks will serve to show that the choice of conversion scheme has significant impact on down-stream performance.

We used the most recent release of the Mate parser first described in Bohnet [20],[5] trained on Sections 2–21 of the Wall Street Journal section of the English Treebank [112]. The graph-based parser is similar to, except much faster, and performs slightly better than the MSTParser [118], which is known to perform well on long-distance dependencies often important for down-stream applications [117, 58, 12]. This choice may of course have an effect on what conversion schemes seem superior [80]. Sentence splitting was done using splitta,[6], and the sentences were then tokenized using PTB-style tokenization[7] and tagged using the in-built Mate POS tagger.

Previous work

There has been considerable work on down-stream evaluation of syntactic parsers in the literature, but most previous work has focused on evaluating parsing models rather than linguistic theories. No one has, to the best of our knowledge, compared the impact of choice of tree-to-dependency conversion scheme across several NLP tasks.

Johansson and Nugues [80] compare the impact of YAMADA and LTH on semantic role labeling performance, showing that LTH leads to superior performance.

Miyao et al. [123] measure the impact of syntactic parsers in an information extraction system identifying protein-protein interactions in biomedical research articles. They evaluate dependency parsers, constituent-based parsers and deep parsers.

Miwa et al. [122] evaluate down-stream performance of linguistic representations and parsing models in biomedical event

5 http://code.google.com/p/mate-tools/
6 http://code.google.com/p/splitta/
7 http://www.cis.upenn.edu/~treebank/tokenizer.sed

extraction, but do not evaluate linguistic representations directly, evaluating representations and models jointly.

Bender et al. [12] compare several parsers across linguistic representations on a carefully designed evaluation set of hard, but relatively frequent syntactic constructions. They compare dependency parsers, constituent-based parsers and deep parsers. The authors argue in favor of evaluating parsers on diverse and richly annotated data. Others have discussed various ways of evaluating across annotation guidelines or translating structures to a common format [148, 172].

Hall et al. [66] discuss optimizing parsers for specific down-stream applications, but consider only a single annotation scheme.

Yuret et al. [184] present an overview of the SemEval-2010 Evaluation Exercises on Semantic Evaluation track on recognition textual entailment using dependency parsing. They also compare several parsers using the heuristics of the winning system for inference. While the shared task is an example of down-stream evaluation of dependency parsers, the evaluation examples only cover a subset of the textual entailments relevant for practical applications, and the heuristics used in the experiments assume a fixed set of dependency labels (EWT labels).

Finally, Schwartz et al. [149] compare the above conversion schemes and several combinations thereof in terms of learnability. This is very different from what is done here. While learnability may be a theoretically motivated parameter, our results indicate that learnability and downstream performance do not correlate well.

11.2 APPLICATIONS

Dependency parsing has proven useful for a wide range of NLP applications, including statistical machine translation [58, 183, 52] and sentiment analysis [82, 79]. This section describes the applications and experimental set-ups included in this study.

In the five applications considered below we use syntactic features in slightly different ways. While our statistical machine translation and sentence compression systems use dependency relations as additional information about words and *on a par* with POS, our negation resolution system uses dependency paths, conditioning decisions on both dependency arcs and labels. In perspective classification, we use dependency triples (e.g. SUBJ(John, snore)) as features, while the semantic role labeling system conditions on a lot of information, including the

word form of the head, the dependent and the argument candidates, the concatenation of the dependency labels of the predicate, and the labeled dependency relations between predicate and its head, its arguments, dependents or siblings.

11.2.1 *Negation resolution*

Negation resolution (NR) is the task of finding negation cues, e.g. the word *not*, and determining their *scope*, i.e. the tokens they affect. NR has recently seen considerable interest in the NLP community [126, 177] and was the topic of the 2012 *SEM shared task [125].

The data set used in this work, the Conan Doyle corpus (CD),[8] was released in conjunction with the *SEM shared task. The annotations in CD extend on cues and scopes by introducing annotations for in-scope events that are negated in factual contexts. The following is an example from the corpus showing the annotations for cues (bold), scopes (underlined) and negated events (italicized):

(11.43) Since we have been so **un***fortunate* as to miss him [...]

CD-style scopes can be discontinuous and overlapping. Events are a portion of the scope that is semantically negated, with its truth value reversed by the negation cue.

The NR system used in this work [95], one of the best performing systems in the *SEM shared task, is a CRF model for scope resolution that relies heavily on features extracted from dependency graphs. The feature model contains token distance, direction, n-grams of word forms, lemmas, POS and combinations thereof, as well as the syntactic features presented in Figure 11. The results in our experiments are obtained from configurations that differ only in terms of tree-to-dependency conversions, and are trained on the training set and tested on the development set of CD. Since the negation cue classification component of the system does not rely on dependency features at all, the models are tested using gold cues.

Table 20 shows F_1 scores for scopes, events and full negations, where a true positive correctly assigns both scope tokens and events to the rightful cue. The scores are produced using the evaluation script provided by the *SEM organizers.

8 http://www.clips.ua.ac.be/sem2012-st-neg/data.html

Syntactic	constituent
	dependency relation
	parent head POS
	grand parent head POS
	word form+dependency relation
	POS+dependency relation
Cue-dependent	directed dependency distance
	bidirectional dependency distance
	dependency path
	lexicalized dependency path

Figure 11: Features used to train the conditional random field models

11.2.2 *Semantic role labeling*

Semantic role labeling (SRL) is the attempt to determine semantic predicates in running text and label their arguments with semantic roles. In our experiments we have reproduced the second best-performing system in the CoNLL 2008 shared task in syntactic and semantic parsing [81].[9]

The English training data for the CoNLL 2008 shared task were obtained from PropBank and NomBank. For licensing reasons, we used OntoNotes 4.0, which includes PropBank, but not NomBank. This means that our system is only trained to classify verbal predicates. We used the Clearparser conversion tool[10] to convert the OntoNotes 4.0 and subsequently supplied syntactic dependency trees using our different conversion schemes. We rely on gold standard argument identification and focus solely on the performance metric semantic labeled F1.

11.2.3 *Statistical machine translation*

The effect of the different conversion schemes was also evaluated on SMT. We used the *reordering by parsing* framework described by Elming and Haulrich [52]. This approach integrates a syntactically informed reordering model into a phrase-based SMT system. The model learns to predict the word order of the

9 http://nlp.cs.lth.se/software/semantic_parsing:_propbank_nombank_frames

10 http://code.google.com/p/clearparser/

translation based on source sentence information such as syntactic dependency relations. Syntax-informed SMT is known to be useful for translating between languages with different word orders [58, 183], e.g. English and German.

The baseline SMT system is created as described in the guidelines from the original shared task.[11] Only modifications are that we use truecasing instead of lowercasing and recasing, and allow training sentences of up to 80 words. We used data from the English-German restricted task: ~3M parallel words of news, ~46M parallel words of Europarl, and ~309M words of monolingual Europarl and news. We use newstest2008 for tuning, newstest2009 for development, and newstest2010 for testing. Distortion limit was set to 10, which is also where the baseline system performed best. The phrase table and the lexical reordering model is trained on the union of all parallel data with a max phrase length of 7, and the 5-gram language model is trained on the entire monolingual data set.

We test four different experimental systems that only differ with the baseline in the addition of a syntactically informed reordering model. The baseline system was one of the tied best performing system in the WMT 2011 shared task on this dataset. The four experimental systems have reordering models that are trained on the first 25,000 sentences of the parallel news data that have been parsed with each of the tree-to-dependency conversion schemes. The reordering models condition reordering on the word forms, POS, and syntactic dependency relations of the words to be reordered, as described in Elming and Haulrich [52]. The paper shows that while reordering by parsing leads to significant improvements in standard metrics such as BLEU [133] and METEOR [96], improvements are more spelled out with human judgements. All SMT results reported below are averages based on 5 MERT runs following Clark et al. [33].

11.2.4 *Sentence compression*

Sentence compression is a restricted form of sentence simplification with numerous usages, including text simplification, summarization and recognizing textual entailment. The most commonly used dataset in the literature is the Ziff-Davis corpus.[12] A widely used baseline for sentence compression experi-

11 http://www.statmt.org/wmt11/translation-task.html

12 LDC Catalog No.: LDC93T3A.

	Baseline	YAMADA	CONLL07	EWT	LTH
Number of dep. rel.	–	12	21	47	41
Parsing					
PTB-23 (LAS)	–	88.99	88.52	81.36*	87.52
PTB-23 (UAS)	–	90.21	90.12	84.22*	**90.29**
Negation scope					
Scope F_1	–	**81.27**	80.43	78.70	79.57
Event F_1	–	76.19	72.90	73.15	**76.24**
Full negation F_1	–	**67.94**	63.24	61.60	64.31
Sentence compression					
Full F_1	68.47	**72.07**	64.29	71.56	71.56
Machine translation					
Dev-Meteor	35.80	36.06	36.06	**36.16**	36.08
Test-Meteor	37.25	37.48	37.50	**37.58**	37.51
Dev-BLEU	13.66	**14.14**	14.09	14.04	14.06
Test-BLEU	14.67	15.04	15.04	14.96	**15.11**
Semantic role labeling					
22-gold	–	81.35	83.22	**84.72**	84.01
23-gold	–	79.09	80.85	80.39	**82.01**
22-pred	–	74.41	76.22	**78.29**	66.32
23-pred	–	73.42	74.34	**75.80**	64.06
Perspective classification					
bitterlemons.org	96.08	**97.06**	95.58	96.08	96.57

Table 20: Results. *: Low parsing results on PTB-23 using EWT are explained by changes between the PTB-III and the Ontonotes 4.0 release of the English Treebank.

ments is Knight and Marcu [89], who introduce two models: the noisy-channel model and a decision tree-based model. Both are tree-based methods that find the most likely compressed syntactic tree and outputs the yield of this tree. McDonald [116] instead use syntactic features to directly find the most likely compressed sentence.

Here we learn a discriminative HMM model [34] of sentence compression using MIRA [38], comparable to previously explored models of noun phrase chunking. Our model is thus neither tree-based nor sentence-based. Instead we think of sentence compression as a sequence labeling problem. We compare a model informed by word forms and predicted POS with models also informed by predicted dependency labels. The baseline feature model conditions emission probabilities on word forms and POS using a ± 2 window and combinations thereoff. The augmented syntactic feature model simply adds dependency labels within the same window.

11.2.5 *Perspective classification*

Finally, we include a document classification dataset from Lin and Hauptmann [99].[13] The dataset consists of blog posts posted at bitterlemons.org by Israelis and Palestinians. The bitterlemons.org website is set up to "contribute to mutual understanding through the open exchange of ideas." In the dataset, each blog post is labeled as either Israeli or Palestinian. Our baseline model is just a standard bag-of-words model, and the system adds dependency triplets to the bag-of-words model in a way similar to Joshi and Penstein-Rose [82]. We do not remove stop words, since perspective classification is similar to authorship attribution, where stop words are known to be informative. We evaluate performance doing cross-validation over the official training data, setting the parameters of our learning algorithm for each fold doing cross-validation over the actual training data. We used soft-margin support vector machine learning [36], tuning the kernel (linear or polynomial with degree 3) and $C = \{0.1, 1, 5, 10\}$.

13 https://sites.google.com/site/weihaolinatcmu/data

Reference:	Zum Glück kam ich beim Strassenbahnfahren an die richtige Stelle .
Source:	Luckily , on the way to the tram , I found the right place .
yamada:	Glücklicherweise hat auf dem Weg zur S-Bahn , stellte ich fest , dass der richtige Ort .
conll07:	Glücklicherweise hat auf dem Weg zur S-Bahn , stellte ich fest , dass der richtige Ort .
ewt:	Zum Glück fand ich auf dem Weg zur S-Bahn , am richtigen Platz .
lth:	Zum Glück fand ich auf dem Weg zur S-Bahn , am richtigen Platz .
Baseline:	Zum Glück hat auf dem Weg zur S-Bahn , ich fand den richtigen Platz .

Figure 12: Examples of SMT output.

Original:	* 68000 sweden ab of uppsala , sweden , introduced the teleserve , an integrated answering machine and voice-message handler that links a macintosh to touch-tone phones .
Baseline:	68000 sweden ab introduced the teleserve an integrated answering machine and voice-message handler .
yamada	68000 sweden ab introduced the teleserve integrated answering machine and voice-message handler .
conll07	68000 sweden ab **sweden** introduced the teleserve integrated answering machine and voice-message handler .
ewt	68000 sweden ab introduced the teleserve integrated answering machine and voice-message handler .
lth	68000 sweden ab introduced the teleserve **an** integrated answering machine and voice-message handler .
Human:	68000 sweden ab introduced the teleserve integrated answering machine and voice-message handler .

Figure 13: Examples of sentence compression output.

Our results are presented in Table 20. The parsing results are obtained relying on predicted POS rather than, as often done in the dependency parsing literature, relying on gold-standard POS. Note that they comply with the result in Schwartz et al. [149] that Yamada-Matsumoto-style annotation is more easily learnable.

NEGATION RESOLUTION The negation resolution results are significantly better using syntactic features in YAMADA annotation. It is not surprising that a syntactically oriented conversion scheme performs well in this task. Since Lapponi et al. [95] used Maltparser [130] with the freely available pre-trained parsing model for English,[14] we decided to also run that parser with the gold-standard cues, in addition to Mate. The pre-trained model was trained on Sections 2–21 of the Wall Street Journal section of the English Treebank [112], augmented with 4000 sentences from the QuestionBank,[15] which was converted using the Stanford converter and thus similar to the EWT annotations used here. The results were better than using EWT with Mate trained on Sections 2–21 alone, but worse than the results obtained here with YAMADA conversion scheme. F_1 score on full negation was 66.92%.

MACHINE TRANSLATION The case-sensitive BLEU evaluation of the SMT systems indicates that choice of conversion scheme has no significant impact on overall performance. The difference to the baseline system is significant ($p < 0.01$), showing that the reordering model leads to improvement using any of the schemes. However, the conversion schemes lead to very different translations. This can be seen, for example, by the fact that the relative tree edit distance between translations of different syntactically informed SMT systems is 12% higher than within each system (across different MERT optimizations).

The reordering approach puts a lot of weight on the syntactic dependency relations. As a consequence, the number of relation types used in the conversion schemes proves important. Consider the example in Figure 12. German requires the verb in second position, which is obeyed in the much better translations produced by the EWT and LTH systems. Interest-

14 http://www.maltparser.org/mco/english_parser/engmalt.html
15 http://www.computing.dcu.ie/~jjudge/qtreebank/

ingly, the four schemes produce virtually identical structures for the source sentence, but they differ in their labeling. Where CONLL07 and YAMADA use the same relation for the first two constituents (ADV and vMOD, respectively), EWT and LTH distinguish between them (ADVMOD/PREP and ADV/LOC). This distinction may be what enables the better translation, since the model may learn to move the verb after the sentence adverbial. In the other schemes, sentence adverbials are not distinguished from locational adverbials. Generally, EWT and LTH have more than twice as many relation types as the other schemes.

SEMANTIC ROLE LABELING The schemes EWT and LTH lead to better SRL performance than CONLL07 and YAMADA when relying on gold-standard syntactic dependency trees. This supports the claims put forward in Johansson and Nugues [80]. These annotations also happen to use a larger set of dependency labels, however, and syntactic structures may be harder to reconstruct, as reflected by labeled attachment scores (LAS) in syntactic parsing. The biggest drop in SRL performance going from gold-standard to predicted syntactic trees is clearly for the LTH scheme, at an average 17.8% absolute loss (YAMADA 5.8%; CONLL07 6.8%; EWT 5.5%; LTH 17.8%).

The EWT scheme resembles LTH in most respects, but in preposition-noun dependencies it marks the preposition as the head rather than the noun. This is an important difference for SRL, because semantic arguments are often nouns embedded in prepositional phrases, like agents in passive constructions. It may also be that the difference in performance is simply explained by the syntactic analysis of prepositional phrases being easier to reconstruct.

SENTENCE COMPRESSION The sentence compression results are generally much better than the models proposed in Knight and Marcu [89]. Their noisy channel model obtains an F_1 compression score of 14.58%, whereas the decision tree-based model obtains an F_1 compression score of 31.71%. While F_1 scores should be complemented by human judgements, as there are typically many good sentence compressions of any source sentence, we believe that error reductions of more than 50% indicate that the models used here (though previously unexplored in the literature) are fully competitive with state-of-the-art models.

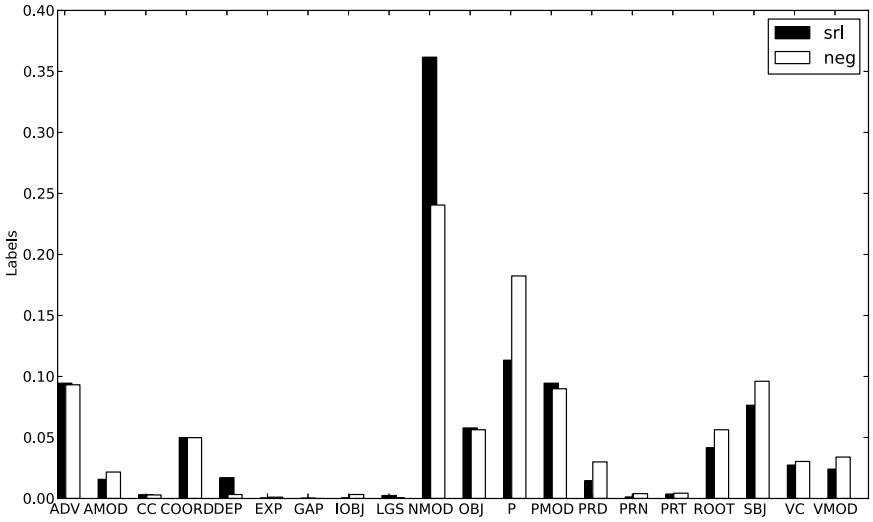


Figure 14: Distributions of dependency labels in the Yamada-Matsumoto scheme

We also see that the models using syntactic features perform better than our baseline model, except for the model using CONLL07 dependency annotation. This may be surprising to some, since distributional information is often considered important in sentence compression [89]. Some output examples are presented in Figure 13. Unsurprisingly, it is seen that the baseline model produces grammatically incorrect output, and that most of our syntactic models correct the error leading to ungrammaticality. The model using EWT annotation is an exception. We also see that CONLL07 introduces another error. We believe that this is due to the way the CONLL07 tree-to-dependency conversion scheme handles coordination. While the word *Sweden* is not coordinated, it occurs in a context, surrounded by commas, that is very similar to coordinated items.

PERSPECTIVE CLASSIFICATION In perspective classification we see that syntactic features based on YAMADA and LTH annotations lead to improvements, with YAMADA leading to slightly better results than LTH. The fact that a syntactically oriented conversion scheme leads to the best results may reflect that perspective classification, like authorship attribution, is less about content than stylistics.

While LTH seems to lead to the overall best results, we stress the fact that the five tasks considered here are incommensurable. What is more interesting is that, task to task, results

are so different. The semantically oriented conversion schemes, EWT and LTH, lead to the best results in SRL, but with a significant drop for LTH when relying on predicted parses, while the YAMADA scheme is competitive in the other four tasks. This may be because distributional information is more important in these tasks than in SRL.

The distribution of dependency labels seems relatively stable across applications, but differences in data may of course also affect the usefulness of different annotations. Note that CONLL07 leads to very good results for negation resolution, but bad results for SRL. See Figure 14 for the distribution of labels in the CONLL07 conversion scheme on the SRL and negation scope resolution data. Many differences relate to differences in sentence length. The negation resolution data is literary text with shorter sentences, which therefore uses more punctuation and has more root dependencies than newspaper articles. On the other hand we do see very few predicate dependencies in the SRL data. This may affect down-stream results when classifying verbal predicates in SRL. We also note that the number of dependency labels have less impact on results in general than we would have expected. The number of dependency labels and the lack of support for some of them may explain the drop with predicted syntactic parses in our SRL results, but generally we obtain our best results with YAMADA and LTH annotations, which have 12 and 41 dependency labels, respectively.

11.4 CONCLUSIONS

We evaluated four different tree-to-dependency conversion schemes, putting more or less emphasis on syntactic or semantic evidence, in five down-stream applications, including SMT and negation resolution. Our results show why it is important to be precise about exactly what tree-to-dependency conversion scheme is used. Tools like pennconverter.jar gives us a wide range of options when converting constituent-based treebanks, and even small differences may have significant impact on down-stream performance. The small differences are also important for more linguistic comparisons that also tend to gloss over exactly what conversion scheme is used, e.g. Ivanova et al. [76].

12

ROBUST LEARNING IN RANDOM SUBSPACES: EQUIPPING NLP FOR OOV EFFECTS

ABSTRACT

Inspired by work on robust optimization we introduce a subspace method for learning linear classifiers for natural language processing that are robust to out-of-vocabulary effects. The method is applicable in live-stream settings where new instances may be sampled from different and possibly also previously unseen domains. In text classification and part-of-speech (POS) tagging, robust perceptrons and robust stochastic gradient descent (SGD) with hinge loss achieve average error reductions of up to 18% when evaluated on out-of-domain data.

12.1 INTRODUCTION

In natural language processing (NLP), data is rarely drawn independently and identically at random. In particular we often apply models learned from available labeled data to data that differs from the original labeled data in several respects. Supervised learning without the assumption that data is drawn identically is sometimes referred to as *transfer learning*, i.e. learning to make predictions about data sampled from a target distribution using labeled data from a *related, but different source distribution* or under a strong *sample bias*.

Domain adaptation refers to a prominent class of transfer learning problems in NLP. Two domain adaptation scenarios are typically considered: (a) *semi-supervised* domain adaption, where a small sample of data from the target domain is available, as well as large pool of unlabeled target data, and (b) *unsupervised* domain adaptation where only unlabeled data is available from the target domain. In this paper we do *not even* assume the latter, but consider the more difficult scenario where the target domain is unknown.

The assumption that a large pool of unlabeled data is available from a relatively homogeneous target domain holds only if the target domain is known in advance. In a lot of applications of NLP, this is not the case. When we design publicly available

software such as the Stanford Parser, or when we set up online services such as Google Translate, we do not know much about the input in advance. A user will apply the Stanford Parser to any kind of text from any textual domain and expect it to do well.[1] Recent work has extended domain adaptation with domain *identification* [48, 115], but this still requires that we know the possible domains in advance and are able to relate each instance to one of them, and in many cases we do not. If the possible target domains are *not* known in advance, the transfer learning problem reduces to the problem of learning robust models that are as insensitive as possible to domain shifts. This is the problem considered in this paper.

One of the main reasons for performance drops when evaluating supervised NLP models on out-of-domain data is out-of-vocabulary (OOV) effects [17, 42]. Several techniques for reducing OOV effects have been introduced in the literature, including spelling expansion, morphological expansion, dictionary term expansion, proper name transliteration, correlation analysis, and word clustering [17, 65, 173, 42], but most of these techniques still leave us with a lot of "empty dimensions", i.e. features that are always 0 in the test data. While these features are not uninstantiated in the sense of missing values, we will nevertheless refer to OOV effects as *removing dimensions* from our datasets, since a subset of dimensions become uninformative as we leave our source domain.

This is a potential source of error, since the best decision boundary in n dimensions is not necessarily the best boundary in $m < n$ dimensions. If we remove dimensions, our optimal decision boundaries may suddenly be far from optimal. Consider, for example, the plot in Figure 15. 2D-SVC is the optimal decision boundary for this two-dimensional dataset (the non-horizontal, solid line). If we remove one dimension, however, say because this variable is never instantiated in our test data, the learned weight vector will give us the decision boundary TEST(2D-SVC) (the dashed line). Compare this to the optimal decision boundary for the reduced, one-dimensional dataset, 1D-SVC (the horizontal, solid line).

OOV effects "remove" dimensions from our data. In robust learning, we do not know which dimensions are to be removed in our target data in advance, however. In this paper we therefore, inspired by previous work on robust optimization [11],

1 Chris Manning previously raised this point in an invited talk at a NAACL workshop.

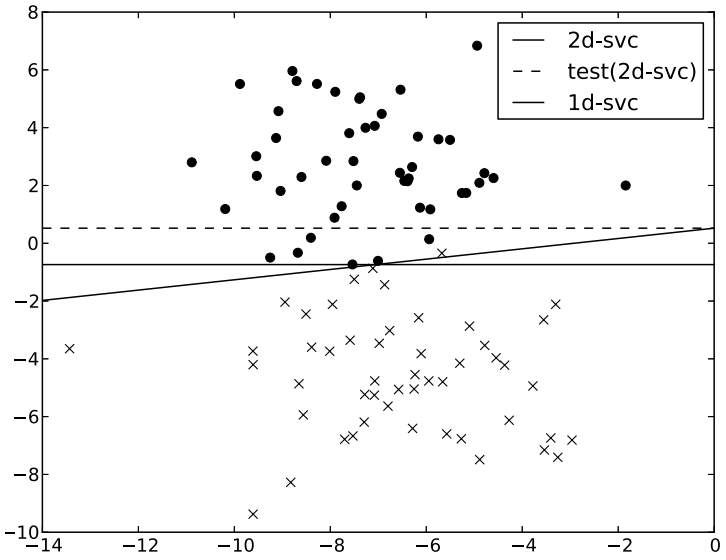


Figure 15: Optimal decision boundary is not optimal when one dimension is removed

suggest to minimize our expected loss under all (or K random) possible removals. We will implement this strategy for perceptron learning and SGD with hinge loss and apply it to text classification, as well as POS tagging. Results are very promising, with error reductions up to 70% and average error reductions up to 18%.

12.2 ROBUST LEARNING UNDER RANDOM SUBSPACES

In robust optimization [11] we aim to find a solution $\mathbf{w}$ that minimizes a (parameterized) cost function $f(\mathbf{w}, \xi)$, where the true parameter ξ may differ from the observed $\hat{\xi}$. The task is to solve

$$\min_{\mathbf{w}} \max_{\hat{\xi} \in \Delta} f(\mathbf{w}, \hat{\xi}) \tag{1}$$

with Δ all possible realizations of ξ. An alternative to minimizing loss in the worst case is minimizing loss in the average case, or the sum of losses:

$$\min_{\mathbf{w}} \sum_{\hat{\xi} \in \Delta} f(\mathbf{w}, \hat{\xi}) \tag{2}$$

The learning algorithms considered in this paper aim to learn models $\mathbf{w}$ from finite samples (of size N) that minimize the expected loss on a distribution ρ (with, say, M dimensions):

$$\min_{\mathbf{w}} \mathbb{E}_{\langle y,\mathbf{x}\rangle\sim\rho} L(y, \text{sign}(\mathbf{w}\cdot\mathbf{x})) \tag{3}$$

OOV effects can be seen as introducing an extra parameter into this equation. Let ξ be a binary vector of length M selecting what dimensions are removed. In NLP we typically assume that $\xi = \langle 1, \ldots, 1\rangle$ and minimize the expected loss in the usual way, but if we have a set Δ of possible instantiations of ξ such that ξ can be any binary vector, minimizing expected loss is likely to be suboptimal, as discussed in the introduction. In this paper we will instead minimize average expected loss *under random subspaces*:

$$\min_{\mathbf{w}} \sum_{\hat{\xi}\in\Delta} \mathbb{E}_{\langle y,\mathbf{x}\rangle\sim\rho} L(y, \text{sign}(\mathbf{w}\cdot\mathbf{x}\circ\hat{\xi})) \tag{4}$$

We refer to this idea as robust learning in random subspaces (RLRS). Since the number of possible instantiations of ξ is 2^M we randomly sample K instantiations removing 10% of the dimensions, with $K \leqslant 250$.[2]

Algorithm 2 Robust learning in random subspaces

```
X = {⟨y_i, x_i⟩}_{i=1}^N
for k ∈ K do
    w^0 = 0, v = 0, i = 0
    ξ ← random.bits(M)
    for n ∈ N do
        if sign(w · x ∘ ξ) ≠ y_n then
            w^{i+1} ← update(w^i)
            i ← i + 1
        end if
    end for
    v ← v + w^i
end for
return w = v/(N × K)
```

2 Our choice to constrain ourselves to instantiations of ξ removing 10% of the dimensions was somewhat arbitrary, and we briefly discuss the effect of this hyper-parameter after presenting our main results.

RLRS can be applied to any linear model, and we present the general form in Figure 2. Given a dataset $X = \{\langle y_i, \mathbf{x}_i \rangle\}_{i=1}^N$ we randomly draw ξ from the set of binary vectors of length M. We now pass over $\{\langle y_i, \mathbf{x}_i \circ \xi \rangle\}_{i=1}^N$ K times, updating our linear model according to the learning algorithm. The weights of the K models are averaged to minimize the average expected loss in random subspaces. In our experiments we will use perceptron [146] and SGD with hinge loss [185] as our learning algorithms. A perceptron c consists of a weight vector $\mathbf{w}$ with a weight for each feature, a bias term b and a learning rate α. For a data point $\mathbf{x}_j$, $c(\mathbf{x}_j) = 1$ iff $\mathbf{w} \cdot \mathbf{x} + b > 0$, else 0. The threshold for classifying something as positive is thus $-b$. The bias term is left out by adding an extra variable to our data with fixed value -1. The perceptron learning algorithm now works by maintaining $\mathbf{w}$ in several passes over the data (see Figure 2). Say the algorithm at time i is presented with a labeled data point $\langle \mathbf{x}_j, y_j \rangle$. The current weight vector $\mathbf{w}^i$ is used to calculate $\mathbf{x}_j \cdot \mathbf{w}^i$. If the prediction is wrong, an update occurs:

$$\mathbf{w}^{i+1} \leftarrow \mathbf{w}^i + \alpha(y_j - \text{sign}(\mathbf{w}^i \cdot \mathbf{x}_j))\mathbf{x}_j \tag{5}$$

The numbers of passes K the learning algorithm does (if it does not arrive at a perfect separator any earlier) is typically fixed by a hyper-parameter. The number of passes is fixed to 5 in our experiments below. The RLRS variant of the perceptron (P-RLRS) is obtained by replacing line 8 in Figure 2 with Equation 5. The application of P-RLRS to an artificial two-dimensional dataset in Figure 16 (the solid line) illustrates how P-RLRS can lead to very different decision boundaries than the regular perceptron (the black dashed line) by averaging decision boundaries learned in random subspaces (red dashed lines).

A perceptron finds the vector $\mathbf{w}$ that minimizes the expected loss on training data where the loss function is given by:

$$L(y, \text{sign}(\mathbf{w} \cdot \mathbf{x})) = \max\{0, -y(\mathbf{w} \cdot \mathbf{x})\} \tag{6}$$

which is 0 when y is predicted correctly, and otherwise the confidence in the mis-prediction. This reflects the fact that perceptron learning is conservative and does not update on correctly classified data points. Equation 6 is the hinge loss with

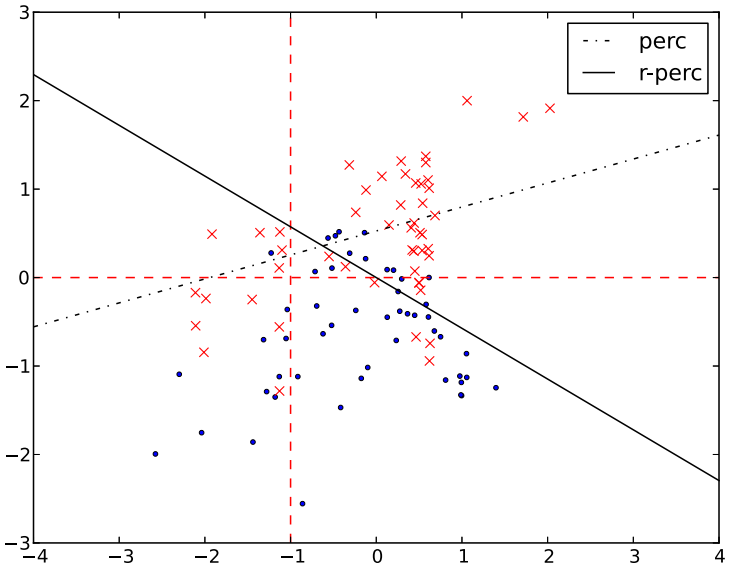


Figure 16: Robust learning in random subspaces (Perceptron on artificial data)

$\gamma = 0$. SGD uses hinge loss with $\gamma = 1$ (like SVMs) [185]. Our objective function thus becomes:

$$\min_{\mathbf{w}} \sum_{\hat{\epsilon} \in \Delta} \mathbb{E}_{\langle y, \mathbf{x} \rangle \sim \rho} \max\{0, \gamma - y(\mathbf{w} \cdot \mathbf{x} \circ \hat{\xi})\} \tag{7}$$

with $\gamma = 0$ for the perceptron and $\gamma = 1$ for SGD. We call the RLRS variant of SGD SGD-RLRS.

12.3 EVALUATION

In our experiments we use perceptron and SGD with hinge loss, regularized using the L_2-norm. Since we want to demonstrate the general applicability of RLRS, we use the default parameters in a publicly available implementation of both algorithms.[3] Both algorithms do five passes over the data. SGD uses 'optimal' learning rate, and perceptron uses a learning rate of 1.

TEXT CLASSIFICATION The goal of text classification is the automatic assignment of documents into predefined semantic classes. The input is a set of labeled documents $\langle y_1, \mathbf{x}_1 \rangle, \ldots, \langle y_N, \mathbf{x}_N \rangle$, and the task is to learn a function $f : \mathcal{X} \mapsto \mathcal{Y}$ that is able to

3 http://scikit-learn.org/stable/

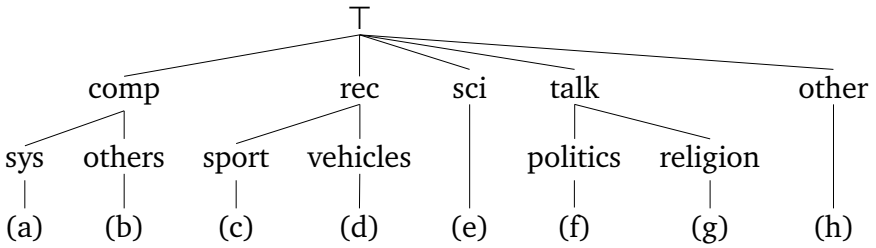


Figure 17: Hierarchical structure of 20 Newsgroups. (a) IBM, Mac, (b) Graphics, MS-Windows, X-Windows, (c) Baseball, Hockey, (d) Autos, Motorcycles, (e) Cryptography, Electronics, Medicine, Space, (f) Guns, Mideast, Miscellaneous, (g) Atheism, Christianity, Miscellaneous, (h) Forsale

correctly classify previously unseen documents. It has previously been noted that robustness is important for the success of text classification in down-stream applications [102]. In this paper we use the 20 Newsgroups dataset.[4] The topics in 20 Newsgroups are hierarchically structured, which enables us to do domain adaptation experiments [29, 160] (except that we will not assume unlabeled data is available in the target domain). See the hierarchy in Figure 17. We extract 20 high-level binary classification problems by considering all pairs of top-level categories, e.g. Computers-Recreative (comp-rec). For each of these 20 problems, we have different possible datasets, e.g. IBM-Baseball, Mac-Motorcycles, etc. A *problem instance* takes training and test data from two *different* datasets belong to the same high-level problem, e.g. Mac-Motorcycles and IBM-Baseball. In total we have 280 available problem instances in the 20 Newsgroups dataset. For each problem instance, we create a sparse matrix of occurrence counts of lowercased tokens and normalize the counts using TF-IDF in the usual way. Otherwise we did not do any preprocessing or feature selection. The code necessary to replicate our text classification experiments is available from the main author's website.[5]

pos tagging To supplement our experiments on the 20 Newsgroups corpus, we also evaluate our approach to robust learning in the context of discriminative HMM training for POS tagging using averaged perceptron [34]. The goal of POS tagging is to assign sequences of labels to words reflecting their syntac-

4 http://people.csail.mit.edu/jrennie/20Newsgroups/
5 http://cst.dk/anders

K	Plain	with RLRS	Error reduction	p-value
Perceptron				
25	67.2	70.1	0.09	< 0.01
50	63.8	66.2	0.07	< 0.01
75	73.2	75.3	0.08	< 0.01
100	72.0	73.3	0.05	~ 0.06
150	72.3	76.2	0.14	< 0.01
250	70.4	72.6	0.07	~ 0.02
SGD				
25	75.2	75.7	0.02	~ 0.17
50	68.6	70.9	0.07	~ 0.02
75	76.3	78.9	0.11	< 0.01
100	73.6	77.1	0.15	< 0.01
150	74.6	79.2	0.18	< 0.01
250	75.0	78.7	0.15	< 0.01

Table 21: Results on 20 Newsgroups

tic categories. We use a publicly available and easy-to-modify reimplementation of the model proposed by Collins (2002).[6] We evaluate our tagger on the English Web Treebank (EWT; LDC2012T13). We use the original PTB tag set, and our results are therefore not comparable to those reported in the SANCL 2012 Shared Task of Parsing the Web. Our model is trained on the WSJ portion of the Ontonotes 4.0 (Sect. 2-21). Our initial experiments used the Email development data, but we simply applied document classification parameters with no tuning. We evaluate our model on test data in the remaining sections of EWT: Answers, Newsgroups, Reviews and Weblogs.

12.3.1 *Results and discussion*

Figure 21 presents our main results on text classification. The left column is the number of extracted subspaces (K in Figure 2). Note that rows are not comparable, since the 20/280 problem instances were randomly selected for each experiment. Neither are the perceptron and SGD results. We observe that P-

6 https://github.com/gracaninja/lxmls-toolkit

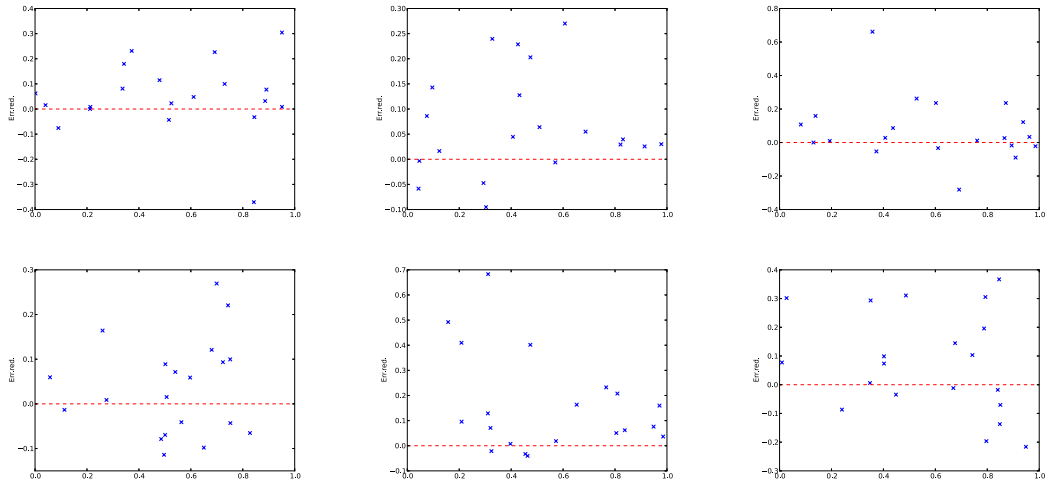


Figure 18: Plots of P-RLRS error reductions with K = 25 (upper left), K = 50 (upper right), K = 75 (lower left), K = 100 (lower mid), K = 150 (lower mid) and K = 250 (lower right).

RLRS consistently outperforms the regular perceptron (P), with error reductions of 7–14%. SGD-RSRL consistently outperforms SGD, with error reductions of 2–18%. Note that statistical significance is *across datasets*, not across data points. Since we are interested in the probability of success on new datasets, we believe this is the right way to evaluate our model, putting our results to a much stronger test. All results, except two, are still statistically significant, however. As one would expect our models become more robust the more instantiations of ξ we sample. The error reductions for each problem instance in the P/P-RLRS experiments are plotted in Figure 18. The plots show that error reductions are up to 70% on some problem instances, and that RLRS seldom hurts (in 3-8 out of 20 cases).

We include a comparison with state-of-the-art learning algorithms for completeness. In Figure 19, we compare SGD-RLRS to passive-aggressive learning (PA) [39] and confidence-weighted learning (CW) [47], using a publicly available implementation,[7] on randomly chosen 20 Newsgroups problem instances. CW is known to be relatively robust to sample bias, reducing weights under-training for correlating features. All algorithms did five passes over the data. Our results indicate that RLRS is more robust than other algorithms, but on some datasets algorithms CW performs much better that RLRS.

7 http://code.google.com/p/oll/ (using default parameters)

	AP	AP-RLRS		
		K = 25	K = 50	K = 100
EWT-Answers	85.22	85.63	**85.69**	85.68
EWT-Newsgroups	86.82	87.26	**87.36**	87.26
EWT-Reviews	84.92	85.32	85.31	**85.35**
EWT-Weblogs	87.00	87.54	87.52	**87.61**

Table 22: Results on the EWT

The results on the EWT are similar to those for 20 Newsgroups, and we observe consistent improvements with both robust averaged perceptron. The results are presented in Table 22. All improvements are statistically significant across data points.

As mentioned, fixing the removal rate to 10% when randomly sampling $\xi \in \Delta$ was a relatively arbitrary choice. RLRS actually benefits slightly from increasing the removal rate. See Figure 20 for results on the selection of problem instances we used in our classifier comparison. In order to explain this we investigated and found a statistically significant correlation between the empirical removal rate and the difference in performance of a model with removal rate 0.8 over a model with removal rate 0.9. This, in our view, suggests that the intuition behind RLRS is correct. Learning under random subspaces *is* a way of equipping NLP for OOV effects.

12.4 RELATED WORK

The RLRS algorithm in Figure 2 is essentially an ensemble learning algorithm, similar in spirit to the random subspace method [72], except averaging over multiple models rather than taking majority votes. Ensemble learning is known to lead to more robust models and therefore to performance gains in domain adaptation [59, 49], so in a way our results are maybe not that surprising. There is also a connection between RLRS and feature bagging [164], a method proposed to reduce weights under-training as an effect of indicative features swamping less indicative features. Weights under-training makes models vulnerable to OOV effects, and feature bagging, in which several models are trained on subsets of features and combined using a mixture of experts, is very similar to RLRS. Sutton et al. 2006 use

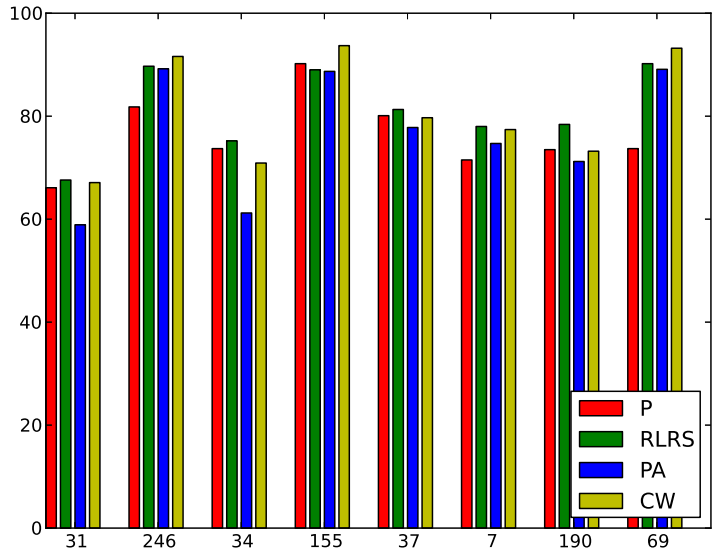


Figure 19: Classifier comparison

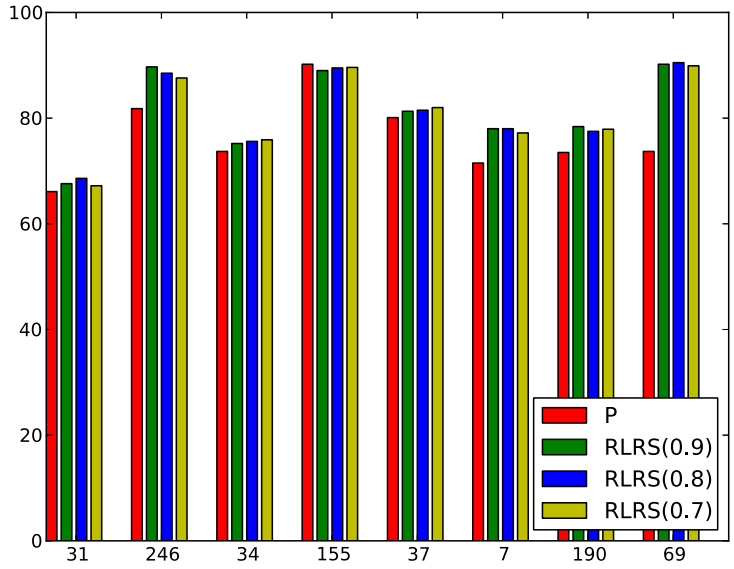


Figure 20: Using increased removal rates when sampling ξ

manually defined rather than random subspaces. See Smith et al. 2005 for an interesting predecessor.

12.5 CONCLUSION

We have presented a novel subspace method for robust learning with applications to document classification and POS tagging, aimed specifically at out-of-vocabulary effects arising in the context of domain adaptation. We have reported average error reductions of up to 18%.

13

FRUSTRATINGLY HARD COMPOSITIONALITY PREDICTION

ABSTRACT

We considered a wide range of features for the DiSCo 2011 shared task about compositionality prediction for word pairs, including COALS-based endocentricity scores, compositionality scores based on distributional clusters, statistics about wordnet-induced paraphrases, hyphenation, and the likelihood of long translation equivalents in other languages. Many of the features we considered correlated significantly with human compositionality scores, but in support vector regression experiments we obtained the best results using only COALS-based endocentricity scores. Our system was nevertheless the best performing system in the shared task, and average error reductions over a simple baseline in cross-validation were 13.7% for English and 50.1% for German.

13.1 INTRODUCTION

The challenge in the DiSCo 2011 shared task is to estimate and predict the semantic compositionality of word pairs. Specifically, the data set consists of adjective-noun, subject-verb and object-verb pairs in English and German. The organizers also provided the Wacky corpora for English and German with lowercased lemmas.[1] In addition, we also experimented with wordnets and using Europarl corpora for the two languages [92], but none of the features based on these resources were used in the final submission.

Semantic compositionality is an ambiguous term in the linguistics litterature. It may refer to the position that the meaning of sentences is built from the meaning of its parts through very general principles of application, as for example in type-logical grammars. It may also just refer to a typically not very well defined measure of semantic transparency of expressions or syntactic constructions, best illustrated by examples:

1 http://wacky.sslmit.unibo.it/

(13.44) pull the plug

(13.45) educate people

The verb-object word pair in Ex. (13.44) is in the training data rated as much less compositional than Ex. (13.45). The intuition is that the meaning of the whole is less related to the meaning of the parts. The compositionality relation is not defined more precisely, however, and this may in part explain why compositionality prediction seems frustratingly hard.

13.2 FEATURES

Many of our features were evaluated with different amounts of *slop*. The slop parameter permits non-exact matches without resorting to language-specific shallow patterns. The words in the compounds are allowed to move around in the sentence one position at a time. The value of the parameter is the maximum number of steps. Set to zero, it is equivalent to an exact match. Below are a couple of example configurations. Note that in order for w_1 and w_2 to swap positions, we must have slop > 1 since slop=1 would place them on top of each other.

x x w_1 w_2 x x (slop=0)
x x w_1 x w_2 x (slop=1)
x x w_1 x x w_2 (slop=2)
x x w_2 w_1 x x (slop=2)

13.2.1 LEFT-ENDOC, RIGHT-ENDOC *and* DISTR-DIFF

These features measure the endocentricity of a word pair w_1 w_2. The distribution of w_1 is likely to be similar to the distribution of "w_1 w_2" if w_1 is the syntactic head of "w_1 w_2". The same is to be expected for w_2, when w_2 is the head.

Syntactic endocentricity is related to compositionality, but the implication is one-way only. A highly compositional compound is endocentric, but an endocentric compound need not be highly compositional. For example, the distribution of "olive oil", which is endocentric and highly compositional, is very similar to the distribution of "oil", the head word. On the other hand, "golden age" which is ranked as highly *non-compositional* in the training data, is certainly endocentric. The distribution of "golden age" is not very different from that of "age".

We used COALS [145] to calculate word distributions. The COALS algorithm builds a word-to-word semantic space from a corpus. We used the implementation by Jurgens and Stevens Jurgens and Stevens [84], generating the semantic space from the Wacky corpora for English and German with duplicate sentences removed and low-frequency words substituted by dummy symbols. The word pairs have been fed to COALS as compounds that have to be treated as single tokens, and the semantic space has been generated and reduced using singular value decompositon. The vectors for w_1, w_2 and "w_1 w_2" are calculated, and we compute the cosine distance between the semantic space vectors for the word pair and its parts, and between the parts themselves, namely for "w_1 w_2" and w_1, for "w_1 w_2" and w_2, and for w_1 and w_2, say for "olive oil" and "olive", for "olive oil" and "oil", and for "olive" and "oil". Left-Endoc is the cosine distance between the left word and the compound. Right-Endoc is the cosine distance between the right word and the compound. Finally, Distr-Diff is the cosine distance between the two words, w_1 and w_2.

13.2.2 Br-Comp

To accommodate for the weaknesses of syntactic endocentricity features, we also tried introducing compositionality scores based on hierarchical distributional clusters that would model semantic compositionality more directly. The scores are referred to below as Br-Comp (compositionality scores based on Brown clusters), and the intuition behind these scores is that a word pair "w_1 w_2", e.g. "hot dog", is non-compositional if w_1 and w_2 have high collocational strength, but if w_1 is replaced with a different word w_1' with similar distribution, e.g. "warm", then "w_1' w_2" is less collocational. Similarly, if w_2 is replaced with a different word w_2' with similar distribution, e.g. "terrier", then "w_1 w_2'" is also much less collocational than "w_1 w_2".

We first induce a hierarchical clustering of the words in the Wacky corpora $cl : W \rightarrow 2^W$ with W the set of words in our corpora, using publicly available software.[2] Let the collocational strength of the two words w_1 and w_2 be $G^2(w_1, w_2)$. We then

2 http://www.cs.berkeley.edu/~pliang/software/

compute the average collocational strength of distributional clusters, BR-CS (collocational strength of Brown clusters):

$$\text{BR-CS}(w_1, w_2) = \frac{\Sigma^{N}_{x \in cl(w_1), x' \in cl(w_2)} G^2(x, x')}{N}$$

with $N = |cl(w_1)| \times |cl(w_2)|$. We now let

$$\text{BR-COMP}(w_1, w_2) = \frac{\text{BR-CS}(w_1, w_2)}{G^2(w_1, w_2)}$$

.

The Brown clusters were built with $C = 1000$ and a cut-off frequency of 1000. With these settings the number of word types per cluster is quite high, which of course has a detrimental effect on the semantic coherence of the cluster. To counter this we choose to restrict $cl(w)$ and $cl(w')$ to include only the 50 most frequently occurring terms.

13.2.3 PARAPHR

These features have to do with alternative phrasings using synonyms from Princeton WordNet [3] and GermaNet[4]. One word in the compound is held constant while the other is replaced with its synonyms. The intuition is again that non-compositional compounds are much more frequent than any compound that results from replacing one of the constituent words with one of its synonyms. For "hot dog" we thus generate "hot terrier" and "warm dog", but not "warm terrier". Specifically, $\text{PARAPHR}_{\geqslant 100}$ means that at least one of the alternative compounds has a document count of more than 100 in the corpus. PARAPHR_{av} is the average count for all paraphrases, PARAPHR_{sum} is the sum of these counts, and PARAPHR_{rel} is the average count for all paraphrases over the count of the word pair in question.

13.2.4 HYPH

The HYPH features were inspired by Bergsma et al. Bergsma et al. [14]. It was only used for English. Specifically, we used the relative frequency of hyphenated forms as features. For adjective-noun pairs we counted the number of hyphenated occurrences, e.g. "front-page", and divided that number by the

3 http://wordnet.princeton.edu/

4 GermaNet Copyright © 1996, 2008 by University of Tübingen.

number of non-hyphenated occurrences, e.g. "front page". For subject-verb and object-verb pairs, we add *-ing* to the verb, e.g. "information-collecting", and divided the number of such forms with non-hyphenated equivalents, e.g. "information collecting".

13.2.5 Trans-Len

The intuition behind our bilingual features is that non-compositional words typically translate into a single word or must be paraphrased using multiple words (circumlocution or periphrasis). Trans-Len is the probability that the phrase's translation, possibly with intervening articles and markers, is longer than l_{min} and shorter than l_{max}, i.e.:

$$\text{Trans-Len}(w_1, w_2, l_{min}, l_{max}) = \frac{\Sigma_{\tau \in trans(w_1\ w_2), l_1 \leqslant |\tau| \leqslant l_2} P(\sigma|w_1\ w_2)}{\Sigma_{\tau \in trans(w_1\ w_2)} P(\sigma|w_1\ w_2)}$$

We use English and German Europarl [92] to train our translation models. In particular, we use the phrase tables of the Moses PB-SMT system[5] trained on a lemmatized version of the WMT11 parallel corpora for English and German. Below Trans-Len-n will be the probability of the translation of a word pair being n or more words. We also experimented with average translation length as a feature, but this did not correlate well with semantic compositionality.

13.3 correlations

We have introduced five different kinds of features, four of which are supposed to model semantic compositionality directly. For feature selection, we therefore compute the correlation of features with compositionality scores and select features that correlate significantly with compositionality. The features are then used for regression experiments.

13.4 regression experiments

For our regression experiments, we use support vector regression with a high (7) degree kernel. Otherwise we use default

5 http://statmt.org

Feature	ρ	
	English	German
rel-type = ADJ_NN	0.0750	*0.1711
rel-type = V_SUBJ	0.0151	**0.2883
rel-type = V_OBJ	0.0880	0.0825
Left-Endoc	**0.3257	*0.1637
Right-Endoc	**0.3896	0.1379
Distr-Diff	*0.1885	0.1128
Hyph (5)	0.1367	-
Hyph (5) reversed	*0.1829	-
G^2	0.1155	0.0535
Br-CS	*0.1592	0.0242
Br-Comp	0.0292	0.0024
Count (5)	0.0795	*0.1523
$\textsc{Paraphr}_{\geqslant \lvert w_1\ w-2\rvert}$	0.1123	0.1242
$\textsc{Paraphr}_{rel}$ (5)	0.0906	0.0013
$\textsc{Paraphr}_{av}$ (1)	0.1080	0.0743
$\textsc{Paraphr}_{av}$ (5)	0.1313	0.0707
$\textsc{Paraphr}_{sum}$ (1)	0.0496	0.0225
$\textsc{Paraphr}_{\geqslant 100}$ (1)	**0.2434	0.0050
$\textsc{Paraphr}_{\geqslant 100}$ (5)	**0.2277	0.0198
Trans-Len-1	0.0797	0.0509
Trans-Len-2	0.1109	0.0158
Trans-Len-3	0.0935	0.0489
Trans-Len-5	0.0240	0.0632

Figure 21: Correlations. Coefficients marked with * are significant ($p < 0.05$), and coefficients marked with ** are highly significant ($p < 0.01$). We omit features with different slop values if they perform significantly worse than similar features.

parameters of publicly available software.[6] In our experiments, however, we were not able to produce substantially better results than what can be obtained using only the features Left-Endoc and Right-Endoc. In fact, for German using only Left-Endoc gave slightly better results than using both. These features are also those that correlate best with human compositionality scores according to Figure 21. Consequently, we only use these features in our official runs. Our evaluations below are cross-validation results on training and development data using leave-one-out. We compare using only Left-Endoc and Right-Endoc (for English) with using all significant features that seem relatively independent. For English, we used Left-Endoc, Right-Endoc, Distr-Diff, Hyph (5) reversed, Br-CS, Paraphr$_{\geqslant}$100 (1). For German, we used rel-type = ADJ_NN, rel-type=V_SUBJ and Right-Endoc. We only optimized on numeric scores. The submitted coarse-grained scores were obtained using average +/- average deviation.[7]

	English		German	
	dev	test	dev	test
Baseline	18.40		47.12	
all sign. indep.	19.22		**23.02**	
L-End+R-End	**15.89**	16.19	23.51	24.03
err.red (L+R)	0.140		0.50	

13.5 Discussion

Our experiments have shown that the DiSCo 2011 shared task about compositionality prediction was a tough challenge. This may be because of the fine-grained compositionality metric or because of inconsistencies in annotation, but note also that the syntactically oriented features seem to perform a lot better than those trying to single out semantic compositionality from syntactic endocentricity and collocational strength. For example, Left-Endoc, Right-Endoc and Br-CS correlate with compositionality scores, whereas Br-Comp does not, although it is

6 http://www.csie.ntu.edu.tw/~cjlin/libsvm/

7 These thresholds were poorly chosen, by the way. Had we chosen less balanced cut-offs, say 0 and 72, our improved accuracy on coarse-grained scores (59.4) would have been comparable to and slightly better than the best submitted coarse-grained scores (58.5).

	Semantic	Syntactic	Collocation	Score
floppy disk			✓	61
free kick	✓			77
happy birthday		✓	✓	47
large scale		✓	✓	55
old school	✓	✓	✓	37
open source		✓	✓	49
real life		✓		69
small group				91

Figure 22: Subjective judgments about semantic and syntactic markedness and collocational strength.

supposed to model compositionality more directly. Could it perhaps be that annotations reflect syntactic endocentricity or distributional similarity to a high degree, rather than what is typically thought of as semantic compositionality?

Consider a couple of examples of adjective-noun pairs in English in Figure 22 for illustration. These examples are taken from the training data, but we have added our subjective judgments about semantic and syntactic markedness and collocational strength (peaking at G^2 scores). It seems that semantic markedness is less important for scores than syntactic markedness and collocational strength. In particular, the combination of syntactic markedness and collocational strength makes annotators rank word pairs such as *happy birthday* and *open source* as non-compositional, although they seem to be fully compositional from a semantic perspective. This may explain why our COALS-features are so predictive of human compositionality scores, and why G^2 correlates better with these scores than BrComp.

13.6 CONCLUSIONS

In our experiments for the DiSCo 2011 shared task we have considered a wide range of features and showed that some of them correlate significantly and sometimes highly significantly with human compositionality scores. In our regression experiments, however, our best results were obtained with only one

or two COALS-based endocentricity features. We report error reductions of 13.7% for English and 50.1% for German.

14

IS FREQUENCY ALL THERE IS TO SIMPLICITY?

ABSTRACT

Our system breaks down the problem of ranking a list of lexical substitutions according to how simple they are in a given context into a series of pairwise comparisons between candidates. For this we learn a binary classifier. As only very little training data is provided, we describe a procedure for generating artificial unlabeled data from Wordnet and a corpus and approach the classification task as a semi-supervised machine learning problem. We use a co-training procedure that lets each classifier increase the other classifier's training set with selected instances from an unlabeled data set. Our features include n-gram probabilities of candidate and context in a web corpus, distributional differences of candidate in a corpus of "easy" sentences and a corpus of normal sentences, syntactic complexity of documents that are similar to the given context, candidate length, and letter-wise recognizability of candidate as measured by a trigram character language model.

14.1 INTRODUCTION

This paper describes a system for the SemEval 2012 English Lexical Simplification shared task. The task description uses a loose definition of simplicity, defining "simple words" as "words that can be understood by a wide variety of people, including for example people with low literacy levels or some cognitive disability, children, and non-native speakers of English" [156].

14.2 FEATURES

We model simplicity with a range of features divided into six groups. Five of these groups make use of the distributional hypothesis and rely on external corpora. We measure a candidate's distribution in terms of its lexical associations (RI), participation in syntactic structures (SYN), or corpus presence in order to assess its simplicity (NGRAM, SW, CHAR). A single

Feature	r	Feature	r
NGRAM_{sf}	0.33	$\text{RI}_{proto(f)}$	-0.15
NGRAM_{sf+1}	0.27	CHAR_{max}	-0.14
NGRAM_{sf-1}	0.27	$\text{RI}_{orig(l)}$	-0.11
LEN_{sf}	-0.26	LEN_{tokens}	-0.10
LEN_{max}	-0.26	CHAR_{min}	0.10
$\text{RI}_{proto(l)}$	-0.18	SW_{freq}	0.08
SYN_{cn}	-0.17	SW_{LLR}	0.07
SYN_{w}	-0.17	CHAR_{avg}	-0.04
SYN_{cp}	-0.17		

Table 23: Pearson's r correlations. The table shows the three highest correlated features per group, all of which are significant at the $p < 0.01$ level

group, LEN, measures intrinsic aspects of the substitution candidate, such as its length.

The substitution candidate is either an adjective, an adverb, a noun, or a verb, and all candidates within a list share the same part of speech. Because word class might influence simplicity, we allow our model to fit parameters specific to the candidate's part of speech by making a copy of the features for each part of speech which is active only when the candidate is in the given part of speech.

SIMPLE WIKIPEDIA (SW) These two features contain relative frequency counts of the substitution form in Simple English Wikipedia (SW_{freq}), and the log likelihood ratio of finding the word in the simple corpus to finding it in regular Wikipedia (SW_{LLR})[1].

WORD LENGTH (LEN) This set of three features describes the length of the substitution form in characters (LEN_{sf}), the length of the longest token (LEN_{max}), and the length of the substitution form in tokens (LEN_{tokens}). Word length is an integral part of common measures of text complexity, e.g in the English Flesch–Kincaid [88] in the form of syllable count, and in the Scandinavian LIX [16].

1 Wikipedia dump obtained March 27, 2012. Date on the Simple Wikipedia dump is March 22, 2012.

CHARACTER TRIGRAM MODEL (CHAR) These three features approximate the reading difficulty of a word in terms of the probabilities of its forming character trigrams, with special characters to mark word beginning and end. A word with an unusual combination of characters takes longer to read and is perceived as less simple [51].

We calculate the minimum, average, and maximum trigram probability (CHAR_{min}, CHAR_{avg}, and CHAR_{max}).[2]

WEB CORPUS N-GRAM (NGRAM) These 12 features were obtained from a pre-built web-scale language model[3]. Features of the form $\text{NGRAM}_{sf \pm i}$, where $0 < i < 4$, express the probability of seeing the substitution form together with the following (or previous) unigram, bigram, or trigram. NGRAM_{sf} is the probability of substitution form itself, a feature which also is the backbone of our frequency baseline.

RANDOM INDEXING (RI) These four features are obtained from measures taken from a word-to-word distributional semantic model. Random Indexing (RI) was chosen for efficiency reasons [147]. We include features describing the semantic distances between the candidate and the original form (RI_{orig}), and between the candidate and a prototype vector (RI_{proto}). For the distance between candidate and original, we hypothesize that annotators would prefer a synonym closer to the original form. A prototype distributional vector of a set of words is built by summing the individual word vectors, thus obtaining a representation that approximates the behavior of that class overall [174]. Longer distances indicate that the currently examined substitution is far from the shared meaning of all the synonyms, making it a less likely candidate. The features are included for both lemma and surface forms of the words.

SYNTACTIC COMPLEXITY (SYN) These 23 features measure the syntactic complexity of documents where the substitution candidate occurs. We used measures from [107] in which they describe 14 automatic measures of syntactic complexity calculated from frequency counts of 9 types of syntactic structures. This group of syntax-metric scores builds on two ideas.

First, syntactic complexity and word difficulty go together. A sentence with a complicated syntax is more likely to be made

2 Trigram probabilities derived from Google T1 unigram counts.

3 The "jun09/body" trigram model from Microsoft Web N-gram Services.

up of difficult words, and conversely, the probability that a word in a sentence is simple goes up when we know that the syntax of the sentence is uncomplicated. To model this we search for instances of the substitution candidates in the UKWAC corpus[4] and measure the syntactic complexity of the documents where they occur.

Second, the perceived simplicity of a word may change depending on the context. Consider the adjective "frigid", which may be judged to be simpler than "gelid" if referring to temperature, but perhaps less simple than "ice-cold" when characterizing someone's personality. These differences in word sense are taken into account by measuring the similarity between corpus documents and substitution contexts and use these values to provide a weighted average of the syntactic complexity measures.

14.3 UNLABELED DATA

The unlabeled data set was generated by a three-step procedure involving synonyms extracted from Wordnet[5] and sentences from the UKWAC corpus.

1. **Collection**: Find synsets for unambigious lemmas in Wordnet. The synsets must have more than three synonyms. Search for the lemmas in the corpus. Generate unlabeled instances by replacing the lemma with each of its synonyms.

2. **Sampling**: In the unlabeled corpus, reduce the number of ranking problems per lemma to a maximum of 10. Sample from this pool while maintaining a distribution of part of speech similar to that of the trial and test set.

3. **Filtering**: Remove instances for which there are missing values in our features.

The unlabeled part of our final data set contains $n = 1783$ problems.

14.4 RANKING

We are given a number of ranking problems ($n = 300$ in the trial set and $n = 1710$ for the test data). Each of these consists

4 http://wacky.sslmit.unibo.it/
5 http://wordnet.princeton.edu/

of a text extract with a position marked for substitution, and a set of candidate substitutions.

14.4.1 *Linear order*

Let $\mathcal{X}^{(i)}$ be the substitution set for the i-th problem. We can then formalize the ranking problem by assuming that we have access to a set of (weighted) preference judgments, $w(a \prec b)$ for all $a, b \in \mathcal{X}^{(i)}$ such that $w(a \prec b)$ is the value of ranking item a ahead of b. The values are the confidence-weighted pair-wise decisions from our binary classifier. Our goal is then to establish a total order on $\mathcal{X}^{(i)}$ that maximizes the value of the non-violated judgments. This is an instance of the Linear Ordering Problem [113], which is known to be NP-hard. However, with problems of our size (maximum ten items in each ranking), we escape these complexity issues by a very narrow margin—$10! \approx 3.6$ million means that the number of possible orderings is small enough to make it feasible to find the optimal one by exhaustive enumeration of all possibilities.

14.4.2 *Binary classication*

In order to turn our ranking problem into binary classification, we generate a new data set by enumerating all point-wise comparisons within a problem and for each apply a transformation function $\Phi(\mathbf{a}, \mathbf{b}) = \mathbf{a} - \mathbf{b}$. Thus each data point in the new set is the difference between the feature values of two candidates. This enables us to learn a binary classifier for the relation "ranks ahead of".

We use the trial set for labeled training data L and, in a transductive manner, treat the test set as unlabeled data U_{test}. Further, we supplement the pool of unlabeled data with artificially generated instances U_{gen}, such that $U = U_{test} \cup U_{gen}$.

Using a co-training setup [19], we divide our features in two independent sets and train a large margin classifier[6] on each split. The classifiers then provide labels for data in the unlabeled set, adding the k most confidently labeled instances to the training data for the other classifier, an iterative process which continues until there is no unlabeled data left. At the end of the training we have two classifiers. The classification result is

6 Liblinear with L1 penalty and L2 loss. Parameter settings were default. http://www.csie.ntu.edu.tw/~cjlin/liblinear/

a mixture-of-experts: the most confident prediction of the two classifiers. Furthermore, as an upper-bound of the co-training procedure, we define an oracle that returns the correct answer whenever it is given by at least one classifier.

14.4.3 *Ties*

In many cases we have items a and b that tie—in which case both $a \prec b$ and $b \prec a$ are violated. We deal with these instances by omitting them from the training set and setting $w(a \prec b) = 0$. For the final ranking, our system makes no attempt to produce ties.

14.5 EXPERIMENTS

In our experiments we vary feature-split, size of unlabeled data, and number of iterations. The first feature split, SYN–SW, pooled all syntactic complexity features and Wikipedia-based features in one view, with the remaining feature groups in another view. Our second feature split, SYN–CHAR–LEN, combined the syntactic complexity features with the character trigram language model features and the basic word length features. Both splits produced a pair of classifiers with similar performance—each had an F-score of around .73 and an oracle score of .87 on the trial set on the binary decision problem, and both splits performed equally on the ranking task.

With a large unlabeled data set available, the classifiers can avoid picking and labeling data points with a low certainty, at least initially. The assumption is that this will give us a higher quality training set. However, as can be seen in Figure 23, none of our systems are benefitting from the additional data. In fact, the systems learn more when the pool of unlabeled data is restricted to the test set.

Our submitted systems, ORD1 and ORD2 scored 0.405 and 0.393 on the test set, and 0.494 and 0.500 on the trial set. Following submission we adjusted a parameter[7] and re-ran each split with both U and U_{test}.

We analyzed the performance by part of speech and compared them to the frequency baseline as shown in Table 24. For the frequency baseline, performance is better on adverbs and

7 In particular, we selected a larger value for the C parameter in the liblinear classifier.

System	All	N	V	R	A
MicrosoftFreq	.449	.367	.456	.487	.493
Syn–SW					
First round	.377	.283	.269	.271	.421
Last round	.425	.355	**.497**	.408	.425
Syn–Char–Len					
First round	.377	.284	.469	.270	.421
Last round	.435	.362	.481	.465	.439

Table 24: Performance on part of speech. Unlabeled set was U_{test}.

adjectives alone, and somewhat worse on nouns. Both our systems benefit from co-training on all word classes. Syn–Char–Len, our best performing system, notably has a score reduction (compared to the baseline) of only 5% on adverbs, eliminates the score reduction on nouns, and effectively beats the baseline score on verbs with a 6% increase.

14.6 discussion

The frequency baseline has proven very strong, and, as witnessed by the correlations in Table 23, frequency is by far the most powerful signal for "simplicity". But is that all there is to simplicity? Perhaps it is. For a person with normal reading ability, a simple word may be just a word with which the person is well-acquainted—one that he has seen before enough times to have a good idea about what it means and in which contexts it is typically used. And so an n-gram model might be a fair approximation. However, lexical simplicity in English may still be something very different to readers with low literacy. For instance, the highly complex letter-to-sound mapping rules are likely to prevent such readers from arriving at the correct pronunciation of unseen words and thus frequent words with exceptional spelling patterns may not seem simple at all.

A source of misclassifications discovered in our error analysis is the fact that substituting candidates into the given contexts in a straight-forward manner can introduce syntactic errors. Fixing these can require significant revisions of the sentence, and yet the substitutions resulting in an ungrammatical sentence are sometimes still preferred to grammatical alterna-

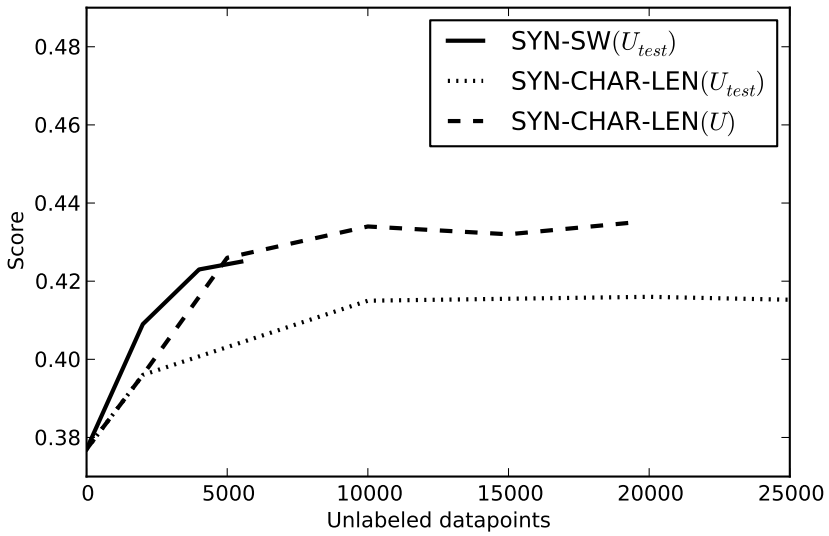


Figure 23: Test set kappa score vs. number of data points labeled during co-training

tives.[8] Here, scoring the substitution and the immediate context in a language model is of little use. Moreover, while these odd grammatical errors may be preferable to many non-native English speakers with adequate reading skills, such errors can be more obstructing to reading impaired users and beginning language learners.

ACKNOWLEDGMENTS

This research is partially funded by the European Commissions 7th Framework Program under grant agreement no. 238405 (CLARA).

8 For example sentence 1528: "However, it appears they intend to *pull* out all stops to get what they want." Gold: {try everything} {do everything it takes} {pull} {stop at nothing} {go to any length} {yank}.

15

CONCLUSION

In this thesis, we have been concerned with answers that differ not in whether they provide correct facts but rather in how these facts are presented. The problem, in other words, has been to automatically identify the answer that provides the best explanation.

We addressed this problem from two different angles:

1. as an answer-ranking problem, learning from community-generated Q&A data; and
2. as a problem of finding adequate representations of answer structure.

To rank answers we learned from user-generated content and ratings collected at community Q&A sites. However, strong biases in the data made it complicated to use. We therefore proposed a method to avoid biasing in favor of early answers but reported a negative result with respect to a bias where longer answers are preferred. In cross-domain answer ranking, we reported error reductions of 20% when training data was sampled according to question similarity.

Our concern with answer structure led to work on disambiguating and classifying the sense of discourse markers. In contrast to previous work on the task, we let go of assumptions about availability of a) gold-standard annotations, and b) labeled examples for all types of discourse markers. Exploring these more realistic evaluation settings resulted in a) more robust models performing at state-of-the-art level, and b) a feature-sharing approach for discourse markers based on syntactical similarity, reducing errors with 20% compared with no sharing.

In a complimentary line of work, we derived more complex answer representations from crowdsourced input. These representations proved effective in reducing errors in answer ranking with 24% compared to a bag-of-words model.

While our focus in the experimental work of the thesis has been on *recognizing* good answers, a challenge of future work is how to *compose* good answers. As discussed in Chapter 5,

query-focused summarization and question answering partly overlap in goals, making it a definite possibility to use a summarizer to generate good answers. This could be accomplished, for instance, by combing a text quality objective with the content selection objective of the summarizer (See Appendix B). We expect models of answer goodness learned on cQA data, like those discussed in the thesis, to be useful for this purpose.

Part III

APPENDIX

A

SAMPLE OF CQA TITLES

A sample of 100 titles from cQA sites in the Stack Exchange network. See 3.2.

Question	Type
Commonly Used Hidden Lists or Objects	Object
Control command arguments	Action
Best way to serve static resoursce -LRB- CSS , Images -RRB- with XDV in Plone	Manner
Print $ Messages in node .	Action
Best way to run PHP with Nginx	Manner
MSDN example scenario	Other
How you would name a .	Manner
Could not load type ‘ site .	Symptom
Implementing join function in a user level thread library	Action
Copy DVD to iTunes for watching on Apple TV	Action
Blocking RDP connections from secondary IP addresses	Action
Yahoo Mail now pops up an “ Add Requests ” tab when I log in .	Symptom
how to determine drive times like those available in google maps	Manner
Resize integral evaluation limits	Action
Appropriate defense for 404s in my logs - persistent web scans from one region	Object
asp.net mvc template missing	Symptom
what knowledge would I need to make a good simulation games	Interrogative
Graphical Android game : Bad performance in some situations	Symptom
Latex in Blogger	Other
Office design and layout for agile development	Other
Toaster Oven pan Without The Toaster Oven	Other
Invensense IMU3000 with PIC	Object
Electron transitions in an infinite square well	Object
How to prepare shallot greens	Manner
computing Impedance related with Voltage Vx	Action
Precautions making carpaccio	Other
tool or technique to get a diff of two different linux installations	Other
incoming mail just sits in the drop folder	Symptom
Connect a List -LRB- Calendar , Task -RRB- with Outlook Results in an Outlook Error 0x8000FFFF	Action

Online notebook , accessible with a userpassword , even from my phone	Other
Execute command on shared account login	Action
How to display one of Drupal 's default forms	Manner
Set a default text format per content type in Drupal 7	Action
Out of the two sql queries below , suggest which one is better one .	Other
MS KB311765 and DsoFramer are missing from MS site	Symptom
Elevating Windows installer in Vista	Action
Adding uploaded images to editor from metabox instead of default popup uploader	Action
Compare content of databases in Oracle	Action
Path separator for Windows and Unix	Object
Retrieving images from a NextGEN gallery	Action
Tangents to a circle from a point outside of it -LRB- tikz -RRB-	Object
Perspective in early pseudo-3d games	Object
Arduino with cell phone	Object
source code check in validation best practices	Other
Should I use the built-in membership provider for an ASP .	Interrogative
Sharepoint application with offline support	Object
MVC Implementation of OAuthConsumer in DotNetOpenAuth specially for Google Address Book	Object
Measuring low ripple on a power supply with an oscilloscope	Action
What are the implications of using .	Interrogative
Possible to add another setting to ' Front page displays ' setting for Custom Post Type	Other
external website storage	Object
Set Time Zone for all Windows Servers on a Domain -LRB- 2008 R2 -RRB-	Action
How to center a quick release rear wheel regularly	Manner
goo .	Other
Parsing optional macro arguments	Action
Do OS X Lion 's Versions and Resume features store the cached data for encrypted .	Interrogative
How to add a WYSIWYG text editor to the Category Edit Screen	Manner
SharePoint 2010 Code : Get list items of list in other site collection	Action
Looking for microcontroller for computer project	Other
Use testdisk and gpart information to mount ext4 partition	Action
Using parallel on Ubuntu	Action
NetBackup Multiplexing for Oracle RMAN Backups	Object
MacOS & finder hang to beach ball after a couple of hard resets , wo n't go away	Symptom
How to prevent someone access the database without permit	Manner

ATMEGA168 -LRB- PV 1020 AU ? -RRB-	Object
Avoiding blank line in every node -LRB- tikzpicture -RRB-	Action
Keeping a published module interesting when some players have already played	Action
Protecting DNS entries from duplicate hostnames entering network	Action
Efficiency of wp_options vs a new table	Other
Difference between ' play you ' and ' play with you '	Other
How to avoid " No Data " from Tiled Map Service in Silverlight	Manner
DHCP server identifier and DHCP relay	Object
ESXi NAS configuration	Object
Installing a class	Action
Connect two Arduinos via simple Serial connection	Action
Mono book recommendations	Other
Energy conservation and quantum measurement	Object
Change Doctype for one Site	Action
Recommend Video Series for Android Development	Other
Book has spacegates that a person can walk across	Other
sorting the linked list	Action
Is my Contact 's birthday in next 10 days	Interrogative
" Anxious to " versus " eager to "	Other
what 's the best way writing php mysql open and close connection	Interrogative
Redgate SQL Compare vs Visual Studio 2010 PremiumUltimate database project	Other
Advice on new hardware firewall for a small company server-environment	Other
How to do a numeric UNIX 's sort on fields with a character attached in front of the number	Manner
hard crack candy coming out too sticky	Symptom
arduino 3x3 LED matrix	Object
Cant find dofollow or no follow in my blog .	Symptom
Resource files creates unnecessary ULS log entries	Symptom
How to have overlapping under-braces and over-braces	Manner
Call Encryption	Object
Horizontal growth vs vertical growth	Other
Upgrading Xserve hard drives	Action
Transform SPOT5 images to natural color images	Action
Grab certain contents of a file	Action
Where should I start and how to progress when learning Java EE	Interrogative
Creating points on multiple lines	Action
How to run regular programs as daemonsservices	Interrogative

B

A CASE STUDY OF A SUMMARIZER

We now take a detailed look at the state-of-the-art summarization system described in Berg-Kirkpatrick et al. [13], which extends [62]. The paper reports the best published ROUGE score on TAC 2008 test data, which is evidence of very good content selection, as ROUGE correlates well with human judgements with respect to content. However, ROUGE is a poor judge of grammaticality and coherence [103], and the system does not explicitly optimize either. On average, the linguistic quality of the summaries is rated as 6.6 out of 10 by workers on Amazon Mechanical Turk, leaving room for improvement. We therefore sketch a way to augment the system using dual decomposition so that it optimizes content and linguistic quality at the same time.

AVOIDING REDUNDANCY

A summary should include the most important information of the source documents, and it should not include it more than once. The naive approach of selecting the most important sentences in order will generally yield a redundant summary, particularly in multi-document summarization where the top-rated sentence is likely to be very similar to sentence two etc., because the same events are mentioned multiple times across the document collection.

Recently a number of promising algorithms have been suggested that jointly address the problem of selecting content and making sure it is non-redundant [120, 50, 142, 26]. Berg-Kirkpatrick et al. [13] deal with the problem in an indirect way by forcing the summarizer to pack as much information as possible into the limited space alloted for the summary. This gives rise to the optimization problem in Eq. (8).

$$\arg\max_{z} \sum_{c \in C(z)} v_c \tag{8}$$

$C(z)$ is the set of information pieces or *concepts* associated with a selection of sentences z, and v_c is the value of including a particular concept c. The objective puts an implicit penalty on

redundancy, because including redundant concepts takes up space while not increasing the value of the objective – each concept counts only once. A concept is a "meaning unit" realized by a sentence and could be anything from words to semantic relations. In the paper, concepts are simply bigrams. The value v_c associated with each concept is estimated from training data.

SOLVING THE OPTIMIZATION PROBLEM

The optimization problem in Eq. (8) can be cast as the following *integer linear program* (ILP):

$$\begin{aligned} \text{maximize} \quad & \sum_c v_c z_c \\ \text{such that} \quad & \sum_s l_s y_s \leqslant L_{max} && (9) \\ \forall\, s, c \quad & y_s Q_{sc} \leqslant z_c && (10) \\ \forall\, c \quad & \sum_s y_s Q_{sc} \geqslant z_c && (11) \end{aligned}$$

z_c and y_s are binary decision variables for concepts and sentences. If a decision variable is on, the corresponding concept or sentence is selected for the summary. Q_{sc} indicates whether concept c is present in sentence s. Eq. (9) is a constraint on the length of the summary which says it cannot exceed L_{max}, with l_s being the length of sentence s. The ILP has two additional constraints stating that

- selecting a sentence implies selecting its concepts [Eq. (10)]
- selecting a concepts implies that at least one sentence containing it must be selected [Eq. (11)].

LINGUISTIC QUALITY AND CONTENT SELECTION

We now take one step back and define the problem of finding the best summary in more abstract terms. Let y represent a summary and $h(y)$ some objective function measuring the goodness of the given summary. Then finding the best summary y^* is a matter of solving the optimization problem below:

$$y^* = \arg\max_{y \in \mathcal{Y}} h(y) \qquad (12)$$

Since ordering is important for assessing the linguistic quality of a summary, we define y as a sequence of sentences. In con-

trast, the optimization problem of Eq. 8 represented the summary as a bag of sentences.

Optimizing linguistic quality and content selection are objectives at odds with each other. The summary maximizing information content (sum of included concepts) is, in general, not the one which is easiest to read. To account for this we see h as really composed of two sub-objectives: f for text quality, and g for content selection, with the parameter γ trading off between them.

$$h(y) = f(y) + \gamma\ g(y) \tag{13}$$

For g we can use the integer linear program discussed above. The text quality objective f could be implemented by the algorithm in Lapata [94], where they learn to order sentences with features extracted from sentence-to-sentence transitions in a corpus. In the next section we give a dual decomposition algorithm for optimizing Eq. 12

DUAL DECOMPOSITION

As before, we consider the problem of optimizing jointly a text quality objective and a content selection objective in summarization. In the optimization problem below, h is the joint objective (repeated from Eq. 12):

$$y^* = \arg\max_{y \in \mathcal{Y}}\ h(y) \tag{14}$$

y is a representation of a summary that captures the order of sentences. In technical terms, if the input documents contain N distinct sentences, then y is an $N+1$ by $N+1$ matrix encoding the pairwise ordering of sentences. For an individual cell,

$$y_{i,j} = \begin{cases} 1 & \text{if sentence } j \text{ is immediately after } i \\ 0 & \text{otherwise} \end{cases}$$

Note that the number of rows and columns in the solution matrix is $N+1$ because we need a dummy sentence 0 in order to represent the start of the sequence. So if the first sentence in the summary is, say, number 3, we set $y_{0,3} = 1$. See Figure 24 for two full examples.

Recall the joint objective h is composed of two sub-objectives f and l, and a parameter γ which trades off between them:

$$h = f(y) + \gamma\ g(l(y)) \tag{15}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 24: Example of solutions y. Both matrices are zero-based. The left matrix encodes the sequence $< 3, 1, 2 >$, and the right encodes $< 1, 2, 3 >$.

$l(y)$ is a function that translates the sequence-aware solution matrix to a bag-of-sentences indicator vector.

The optimization problem from before, replacing $h(y)$ by Eq. (13),

$$y^* = \arg\max_{y \in \mathcal{Y}} f(y) + \gamma\ g(l(y)), \tag{16}$$

can be rewritten as

$$(y^*, z^*) = \arg\max_{y \in \mathcal{Y},\ z \in \mathcal{Z}} f(y) + \gamma\ g(z), \tag{17}$$

such that $l(y)_i = z_i$ for all $i \in \{1 \dots N\}$. These constraints force agreement between the solutions y and z. Thus the optimal solution y^* of Eq. (16) is the same as the y^* of Eq. (17). To enforce the agreement we introduce a vector of Lagrange multipliers u, where $u_i \in \mathbb{R}$ for all $i \in \{1 \dots N\}$.

The Lagrangian is:

$$L(u, y, z) = \left(f(y) + \sum_i u_i l(y)_i \right) + \gamma \left(g(z) - \sum_i u_i z_i \right) \tag{18}$$

The dual objective then is:

$$L(u) = \max_{y \in \mathcal{Y}, z \in \mathcal{Z}} L(u, y, z) \tag{19}$$

$$= \max_{y \in \mathcal{Y}} \left(f(y) + \sum_i u_i l(y)_i \right) + \max_{z \in \mathcal{Z}} \gamma \left(g(z) - \sum_i u_i z_i \right) \tag{20}$$

Equation (20) shows the decomposition of the original problem into two subproblems. The max of the two terms can be calculated independently and summed.

Now, in order to solve the subproblems, we must incorporate the Lagrange multipliers into each of them. In the case of the content selection objective $g(z)$, it is straight-forward. We simply change the objective function of the ILP to directly optimize Eq. (21).

$$\arg\max_{z \in \mathcal{Z}} \; g(z) - \sum_{i=1}^{N} u_i z_i \tag{21}$$

$$= \arg\max_{z \in \mathcal{Z}} \sum_{c \in C(z)} v_c - \sum_{i=1}^{N} u_i z_i \tag{22}$$

Our task is now to minimize the dual objective $L(u)$. We use a subgradient algorithm, which iteratively updates the Lagrange multipliers according to the agreement between the two solutions.

At the k+1-th iteration the update will be

$$u_i^{(k+1)} = u_i^{(k)} - \delta_k (l(y^{(k)})_i - z_i^{(k)}) \tag{23}$$

δ_k is the update rate for round k. In the cases where the two solutions agree the value will be unchanged. If on the other hand a sentence is selected in the y solution but not in the z solution, the update to the Langrange multiplier at the corresponding index will be $-\delta_k$. Intuitively, the effect is to decrease the value of selecting that sentence in the next iteration for the $f(y)$ objective, penalizing the choice. For the $g(z)$ objective the situation is the opposite: the value of including the sentence in the solution is increased by δ_k.

C

FEATURES TRACKING LENGTH

In the motivating experiment of Section 2, we observed that features corresponding to very frequent tokens (e.g. "was" and "as") were highly predictive of *why* answers. We suggested that the features might in fact be modeling a difference in average length between *how* and *why* answers.

We now explain why this happens, even when feature vectors are normalized to unit length. Without normalization, longer documents have a higher count of common vocabulary items.

The normalization factor is the Euclidean length of the vector, given by

$$\sqrt{v_1^2 + v_2^2 + \ldots + v_n^2},$$

where v_1 to v_n are the observed counts of vocabulary items. When the vocabulary items are Zipf-like distributed, the normalization factor will not grow at the same rate as the largest counts. This means longer documents will have higher normalized values for common words. See also discussion in Singhal et al. [153].

AN EXAMPLE

When documents get longer, the vocabulary increases, and many of the new words appear only once in the document. However, grammatical elements, such as inflections of the word "be", increase in frequency, because they are needed in almost every sentence. We now show how the normalized feature value of the most common vocabulary item varies as a function of document length. Assuming the feature is not implicitly tracking the length of the document, the curve should be flat. However, Figure 25 shows a strong dependece between document size and feature value.

The documents are generated using two sets of assumptions:

- FIXED A single grammatical element makes up 10% of the tokens. The rest of the vocabulary items occur only once. Thus a document with 20 tokens tokens will have two occurrences of the grammatical element and 18 other unique vocabulary elements.

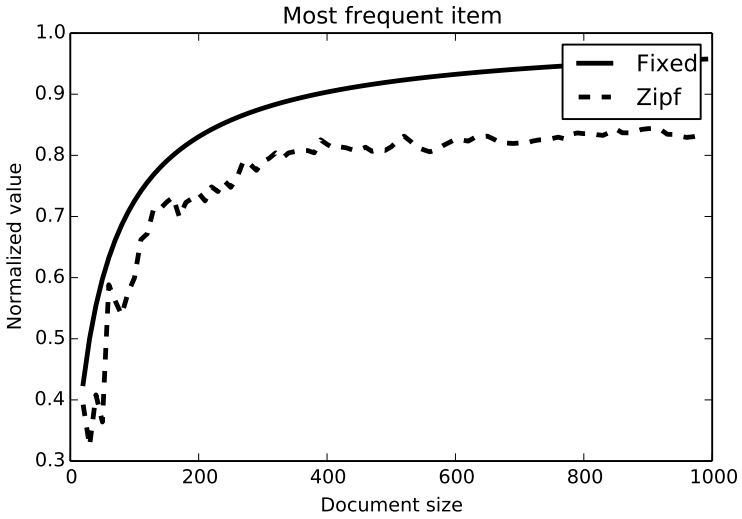


Figure 25: The plot shows normalized feature value as a function of document length in tokens. A flat line would indicate that the feature is not implicitly tracking document length.

- ZIPF The vocabulary follows the Zipf distribution. We assume a vocabulary size of 10,000 and set $\alpha = 1.0$.

BIBLIOGRAPHY

[1] LA Adamic and J Zhang. Knowledge Sharing and Yahoo Answers: Everyone Knows Something. *WWW '08 Proceedings of the 17th international conference on World Wide Web*, 2008. URL http://dl.acm.org/citation.cfm?id=1367587.

[2] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the international conference on Web search and web data mining*, pages 183–194. ACM, 2008.

[3] Naoyoshi Aikawa, Tetsuya Sakai, and Hayato Yamana. Community QA Question Classification: Is the Asker Looking for Subjective Answers or Not? *IPSJ Online Transactions*, 4:160–168, 2011. ISSN 1882-6660. doi: 10.2197/ipsjtrans.4.160. URL http://www.citeulike.org/user/chaozhou/article/9605775.

[4] Ablimit Aji and Eugene Agichtein. The nays have it: exploring effects of sentiment in collaborative knowledge sharing. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, 2010. URL http://dl.acm.org/citation.cfm?id=1860668.

[5] Amal Alsaif and Katja Markert. Modelling discourse relations for Arabic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 736–747, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145517.

[6] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012. URL http://dl.acm.org/citation.cfm?id=2339665.

[7] A Argyriou, A Maurer, and M Pontil. An Algorithm for Transfer Learning in a Heterogeneous Environment. *Machine Learning and Knowledge Discovery in Databases*, 5211: 71–85, 2008.

[8] Michaela Atterer and Hinrich Schütze. Prepositional phrase attachment without oracles. *Computational Linguistics*, 33(4):469–476, 2007.

[9] Regina Barzilay and Mirella Lapata. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34:1–34, 2008. ISSN 08912017. doi: 10.1162/coli.2008.34.1.1.

[10] Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research*, 17:35–55, June 2002. doi: 10.1613/jair.991. URL http://arxiv.org/abs/1106.1820.

[11] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23 (4), 1998.

[12] Emily Bender, Dan Flickinger, Stephan Oepen, and Yi Zhang. Parser evaluation over local and non-local dependencies in a large corpus. In *EMNLP*, 2011.

[13] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly learning to extract and compress. In *Proc. of ACL*, 2011. URL http://www.cs.berkeley.edu/~tberg/papers/acl2011.pdf.

[14] Shane Bergsma, Aditya Bhargava, Hua He, and Grzegorz Kondrak. Predicting the semantic compositionality of prefix verbs. In *EMNLP*, 2010.

[15] J Bian, Y Liu, E Agichtein, and H Zha. Finding the right facts in the crowd: factoid question answering over social media. *WWW '08 Proceedings of the 17th international conference on World Wide Web*, 2008. URL http://dl.acm.org/citation.cfm?id=1367561.

[16] C. H. Bjornsson. Readability of Newspapers in 11 Languages. *Reading Research Quarterly*, 18(4):480–497, 1983. ISSN 00340553. doi: 10.2307/747382. URL http://www.jstor.org/stable/747382?origin=crossref.

[17] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL*, 2007.

[18] MJ Blooma, AYK Chua, and DHL Goh. A predictive framework for retrieving the best answer. *Proceedings of the 2008 ACM symposium on Applied computing*, 2008. URL http://dl.acm.org/citation.cfm?id=1363944.

[19] A Blum and T Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[20] Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*, 2010.

[21] Dwight Le Merton Bolinger. *Interrogative structures of American English (The direct question)*. Published for the Society by University of Alabama Press, 1957. URL http://www.amazon.com/Interrogative-structures-American-English-Publication/dp/B0007ILQYE.

[22] Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daume. Besting the quiz master: Crowdsourcing incremental classification games. In *NAACL*, 2012.

[23] G Burel, Y He, and H Alani. Automatic identification of best answers in online enquiry communities. *The Semantic Web: Research and Applications*, 2012. URL http://link.springer.com/chapter/10.1007/978-3-642-30284-8_41.

[24] Jill Burstein and Martin Chodorow. Progress and New Directions in Technology for Automated Essay Evaluation. In R. Kaplan, editor, *The Oxford Handbook of Applied Linguistics, 2nd Edition*, pages 487–497. Oxford University Press, 2010.

[25] Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. Automated scoring using a hybrid feature identification technique. In *ACL*, 1998.

[26] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '98*, pages 335–336, 1998. doi: 10.1145/290941.291025. URL http://portal.acm.org/citation.cfm?doid=290941.291025.

[27] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Springer, 2003.

[28] Dana R Carney and Mahzarin R Banaji. First Is Best. *PLoS ONE*, 7(6):e35088, June 2012. URL http://dx.doi.org/10.1371/journal.pone.0035088.

[29] Bo Chen, Wai Lam, Ivor Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In *KDD*, 2009.

[30] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *ICML*, 2012.

[31] Yanqing Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. The Expressive Power of Word Embeddings. page 8, January 2013. URL http://arxiv.org/abs/1301.3226.

[32] Erik Choi, Vanessa Kitzie, and Chirag Shah. Developing a typology of online Q&A models and recommending the right model for each question type. In *HCIR 2012*, number 3, 2012. URL http://onlinelibrary.wiley.com/doi/10.1002/meet.14504901302/full.

[33] Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *ACL*, 2011.

[34] Michael Collins. Discriminative training methods for Hidden Markov Models. In *EMNLP*, 2002.

[35] Mike Collins. *Head-driven statistical models for natural language parsing*. PhD thesis, University of Pennsylvania, 1999.

[36] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[37] Stella Cottrell. *The Study Skills Handbook (Palgrave Study Guides)*. Palgrave Macmillan, 2003. URL http://www.amazon.co.uk/Study-Skills-Handbook-Palgrave-Guides/dp/1403911355.

[38] Koby Crammer and Yoram Singer. Ultraconservative algorithms for multiclass problems. In *JMLR*, 2003.

[39] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-agressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[40] David Crystal. Internet Linguistics: A Student Guide. March 2011. URL http://dl.acm.org/citation.cfm?id=2011923.

[41] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a large annotated corpus of learner {E}nglish. In *Workshop on Innovative Use of NLP for Building Educational Applications, NAACL*, 2013.

[42] Hal Daumé and Jagadeesh Jagarlamudi. Domain adaptation for machine translation by mining unseen words. In *ACL*, 2011.

[43] Hal Daumé III. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*, volume 45, page 256, 2007. doi: 10.1.1.110.2062. URL http://acl.ldc.upenn.edu/P/P07/P07-1033.pdf.

[44] Hal Daumé III, Abhishek Kumar, and Avishek Saha. Frustratingly easy semi-supervised domain adaptation. pages 53–59, July 2010. URL http://dl.acm.org/citation.cfm?id=1870526.1870534.

[45] Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. Multi-view learning of word embeddings via cca. In *Advances in Neural Information Processing Systems*, pages 199–207, 2011.

[46] Pedro Domingos. A few useful things to know about machine learning. In *CACM*, 2012.

[47] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *ICML*, 2008.

[48] Mark Dredze, Tim Oates, and Christine Piatko. We're not in {K}ansas anymore: detecting domain changes in streams. In *EMNLP*, 2010.

[49] Lixin Duan, Ivor Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxilliary classifiers. In *ICML*, 2009.

[50] Avinava Dubey and S Chakrabarti. Diversity in ranking via resistive graph centers. In *KDD '11 Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 78–86, 2011. ISBN 9781450308137. URL http://mllab.csa.iisc.ernet.in/html/pubs/p78-dubey.pdf.

[51] Linnea C Ehri. Learning to read words: Theory, findings, and issues. *Scientific Studies of reading*, 9(2):167–188, 2005.

[52] Jakob Elming and Martin Haulrich. Reordering by Parsing. In *Proceedings of International Workshop on Using Linguistic Information for Hybrid Machine Translation (LIHMT-2011)*, 2011.

[53] Micha Elsner and Eugene Charniak. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT, Short Papers*, number June, pages 41–44, Columbus, Ohio, 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-2011.

[54] Pnina Fichman. A comparative assessment of answer quality on four question answering sites. *Journal of Information Science*, 37(5):476–486, 2011.

[55] Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. Annotating named entities in {T}witter data with crowdsourcing. In *NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010.

[56] Bruce Fraser. What are discourse markers? *Journal of Pragmatics*, 31:931–952, 1999. ISSN 03782166. doi: 10.1016/S0378-2166(98)00101-5.

[57] Alice Freed and Susan Ehrlich, editors. *Why Do You Ask?: The Function of Questions in Institutional Discourse*. Oxford University Press, USA, 2010. ISBN 0195306902. URL http://www.amazon.com/Why-You-Ask-Questions-Institutional/dp/B007K5GJ08.

[58] Michel Galley and Christopher Manning. Quadratic-time dependency parsing for machine translation. In *ACL*, 2009.

[59] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *KDD*, 2008.

[60] Sucheta Ghosh. *End-to-End Discourse Parse using Cascaded Structured Prediction*. Phd thesis, University of Trento, 2012.

[61] P Gianfortoni, D Adamson, and CP Rosé. Modeling of stylistic variation in social media with stretchy patterns. *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, 2011. URL http://dl.acm.org/citation.cfm?id=2140539.

[62] Dan Gillick and Benoit Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18. Association for Computational Linguistics, June 2009. ISBN 978-1-932432-35-0. URL http://dl.acm.org/citation.cfm?id=1611638.1611640.

[63] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for Twitter: annotation, features, and experiments. pages 42–47, June 2011. URL http://dl.acm.org/citation.cfm?id=2002736.2002747.

[64] Sidney Greenbaum and Randolph Quirk. *A Student's Grammar of the English Language (Grammar Reference)*. Addison Wesley Publishing Company, 1990. ISBN 0582059712. URL http://www.amazon.com/Students-Grammar-English-Language-Reference/dp/0582059712.

[65] Nizar Habash. Four techniques for online handling of out-of-vocabulary words in {A}rabic-{E}nglish statistical machine translation. In *ACL*, 2008.

[66] Keith Hall, Ryan McDonald, Jason Katz-Brown, and Michael Ringgaard. Training dependency parsers by jointly optimizing multiple objectives. In *EMNLP*, 2011.

[67] Michael A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, 1976. URL http://books.google.com/books?id=AzibnAEACAAJ&pgis=1.

[68] F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan. Predictors of answer quality in online Q&A sites. In *CHI '08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008. URL http://dl.acm.org/citation.cfm?id=1357191.

[69] F. Maxwell Harper, Daniel Moy, and Joseph A. Konstan. Facts or friends? Distinguishing informational and conversational questions in social Q&A sites. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, page 759, New York, New York, USA, April 2009. ACM Press. ISBN 9781605582467. doi: 10.1145/1518701.1518819. URL http://dl.acm.org/citation.cfm?id=1518701.1518819.

[70] Yulan He and Harith Alani. Automatic Identification of Best Answers. In *9th Extended Semantic Web Conference 2012*, pages 514–529, 2012.

[71] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. In *9th International Conference on Artificial Neural Networks: ICANN '99*, volume 1999, pages 97–102. IEE, 1999. ISBN 0 85296 721 7. doi: 10.1049/cp:19991091. URL http://www.mendeley.com/research/support-vector-learning-ordinal-regression/.

[72] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[73] Hen-Hsen Huang and Hsin-Hsi Chen. Chinese discourse relation recognition. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 1442–1446, 2011.

[74] Rodney Huddleston and Geoffrey K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, 2002. ISBN 0521431468. URL http://www.amazon.com/The-Cambridge-Grammar-English-Language/dp/0521431468.

[75] Ben Hutchinson. Modelling the substitutability of discourse connectives. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, pages 149–156, Morristown, NJ, USA, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219859. URL http://dl.acm.org/citation.cfm?id=1219840.1219859.

[76] Angelina Ivanova, Stephan Oepen, Lilja Ø vrelid, and Dan Flickinger. Who did what to whom? A contrastive study of syntactico-semantic dependencies. In *LAW*, 2012.

[77] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. A framework to predict the quality of answers with non-textual features. *SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006. URL http://dl.acm.org/citation.cfm?id=1148212.

[78] Sneha Jha, Hansen A. Schwartz, and Lyle Ungar. Penn: Using Word Similarities to better Estimate Sentence Similarity. In *SEM 2012: The First Joint Conference on Lexical and Computational Semantics Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 679–683, Montréal, 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S12-1101.

[79] Richard Johansson and Alessandro Moschitti. Syntactic and semantic structure for opinion expression detection. In *CoNLL*, 2010.

[80] Richard Johansson and Pierre Nugues. Extended constituent-to-dependency conversion for {E}nglish. In *NODALIDA*, 2007.

[81] Richard Johansson and Pierre Nugues. Dependency-based syntactic-semantic analysis with propbank and nombank. In *CoNLL*, 2008.

[82] Mahesh Joshi and Carolyn Penstein-Rose. Generalizing dependency features for opinion mining. In *ACL*, 2009.

[83] Mahesh Joshi, Mark Dredze, William W Cohen, and Carolyn Rose. Multi-Domain Learning: When Do Domains Matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1302–1312, Jeju Island, Korea, 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D12-1119.

[84] David Jurgens and Keith Stevens. The {S}-{S}pace package: an open source package for word space models. In *ACL*, 2010.

[85] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpus–a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–i182, July 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/btg1023. URL http://bioinformatics.oxfordjournals.org/content/19/suppl_1/i180.short.

[86] S Kim and S Oh. Users' relevance criteria for evaluating answers in a social Q&A site. *Journal of the American Society for Information Science and Technology*, pages 716–727, 2009. URL http://onlinelibrary.wiley.com/doi/10.1002/asi.21026/full.

[87] Soojung Kim, Jung Sun Oh, and Sanghee Oh. Best-Answer Selection Criteria in a Social Q&A site from the User-Oriented Relevance Perspective. In *American Society for Information Science and Technology*, 2007. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.1799.

[88] J P Kincaid, R P Fishburne, R L Rogers, and B S Chissom. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel, 1975. URL http://www.eric.ed.gov/ERICWebPortal/detail?accno=ED108134.

[89] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence*, 139:91–107, 2002.

[90] Alistair Knott. *A data-driven methodology for motivating a set of coherence relations*. PhD thesis, University of Edinburgh, 1996.

[91] Alistair Knott and Robert Dale. Using linguistic phenomena to motivate a set of coherence relations. *Discourse Processes*, 18(1):35–62, July 1994. ISSN 0163-853X. doi: 10.1080/01638539409544883. URL http://dx.doi.org/10.1080/01638539409544883.

[92] Philipp Koehn. Europarl: a parallel corpus for statistical machine translation. In *MT-Summit*, 2005.

[93] Liang-Cheng Lai and Hung-Yu Kao. Question Routing by Modeling User Expertise and Activity in cQA services. In *The 26th Annual Conference of the Japanese Society for Artificial Intelligence*, 2012.

[94] Mirella Lapata. Probabilistic Text Structuring : Experiments with Sentence Ordering. In *Proceedings of the 41st Meeting of the Association of Computational Linguistics*, pages 545–552, 2003.

[95] Emanuele Lapponi, Erik Velldal, Lilja Ø vrelid, and Jonathon Read. {U}i{O}2: Sequence-labeling Negation Using Dependency Features. In *SEM, 2012.

[96] Alon Lavie and Abhaya Agarwal. Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *WMT*, 2007.

[97] Jakob R. E. Leimgruber. *Modelling Variation in Singapore English*. PhD thesis, University of Oxford, 2009.

[98] Jimmy Lin and Dina Demner-Fushman. Evaluating Summaries and Answers: Two Sides of the Same Coin? In *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 41–48, 2005.

[99] Wei-Hao Lin and Alexander Hauptmann. Are these documents written from different perspectives? In *COLING-ACL*, 2006.

[100] Ziheng Lin, Hwee Tou Ng, and Min-yen Kan. A PDTB-Styled End-to-End Discourse Parser. Technical Report 2004, School of Computing, National University of Singapore, Singapore, 2010.

[101] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. Automatically evaluating text coherence using discourse relations. pages 997–1006, June 2011. URL http://dl.acm.org/citation.cfm?id=2002472.2002598.

[102] Nedim Lipka and Benno Stein. Robust models in information retrieval. In *DEXA*, 2011.

[103] Elena Lloret and Manuel Palomar. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, pages 1–41, 2011. ISSN 0269-2821. URL http://dx.doi.org/10.1007/s10462-011-9216-z.

[104] John Logie, Joseph Weinberg, F Maxwell Harper, and Joseph A Konstan. Asked and Answered: On Qualities and Quantities of Answers in Online Q&A Sites. In *The Social Mobile Web*, 2011.

[105] Annie Louis and Ani Nenkova. What Makes Writing Great? First Experiments on Article Quality Prediction in the Science Journalism Domain. *Transactions of the ACL*, 2013.

[106] Annie Louis, Aravind Joshi, and Ani Nenkova. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, pages 147–156, Tokyo, Japan, September 2010. Association for Computational Linguistics, Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W10/W10-4327.

[107] Xiaofei Lu. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15(4):474–496, 2010.

[108] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[109] Christopher D. Manning. Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In Alexan-

derF. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing SE - 14*, volume 6608 of *Lecture Notes in Computer Science*, pages 171–189. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19399-6. doi: 10.1007/978-3-642-19400-9_14. URL http://dx.doi.org/10.1007/978-3-642-19400-9_14.

[110] Daniel Marcu. Discourse Trees Are Good Indicators of Importance in Text. In *Advances in Automatic Text Summarization*, 1999. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.8292.

[111] Daniel Marcu and Abdessamad Echihabi. An Unsupervised Approach to Recognizing Discourse Relations. In *Proc. of ACL*, 2002. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.669.

[112] Mitchell Marcus, Mary Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of {E}nglish: the {P}enn {T}reebank. *Computational Linguistics*, 19(2): 313–330, 1993.

[113] Rafael Martí and Gerhard Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization (Applied Mathematical Sciences)*. Springer, 2011. ISBN 3642167284. URL http://www.amazon.com/Linear-Ordering-Problem-Combinatorial-Optimization/dp/3642167284.

[114] Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *ICDM*, 2012.

[115] David McClosky, Eugene Charniak, and Mark Johnson. Automatic domain adaptation for parsing. In *NAACL-HLT*, 2010.

[116] Ryan McDonald. Discriminative sentence compression with soft syntactic evidence. In *EACL*, 2006.

[117] Ryan McDonald and Joakim Nivre. Characterizing the errors of data-driven dependency parsers. In *EMNLP-CoNLL*, 2007.

[118] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on*

Human Language Technology and Empirical Methods in Natural Language Processing 2005, pages 523–530, Vancouver, British Columbia, 2005.

[119] G Harry McLaughlin. SMOG Grading a New Readability Formula. *Journal Of Reading*, 12(8):639–646, 1969.

[120] Qiaozhu Mei and Jian Guo. Divrank: the interplay of prestige and diversity in information networks. In *KDD '10 Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1009–1018, 2010. ISBN 9781450300551. URL http://portal.acm.org/citation.cfm?id=1835931.

[121] Thomas Meyer and Andrei Popescu-Belis. Using Sense-labeled Discourse Connectives for Statistical Machine Translation. In *Proceedings of the Joint Workshop on Exploiting Synergies Between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, EACL 2012, pages 129–138, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-19-0. URL http://dl.acm.org/citation.cfm?id=2387956.2387973.

[122] Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. Evaluating dependency representation for event extraction. In *COLING*, 2010.

[123] Yusuke Miyao, Rune Sæ tre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. Task-oriented evaluation of syntactic parsers and their representations. In *ACL*, 2008.

[124] Yoko Mizuta, Anna Korhonen, Tony Mullen, and Nigel Collier. Zone analysis in biology articles as a basis for information extraction. *International Journal of Medical Informatics*, 75:468–487, 2006.

[125] Roser Morante and Eduardo Blanco. *SEM 2012 Shared Task: Resolving the Scope and Focus of Negation. In **SEM*, 2012.

[126] Roser Morante and Caroline Sporleder. Modality and Negation: An Introduction to the Special Issue. *Computational linguistics*, 38(2):223–260, 2012.

[127] Alessandro Moschitti and Silvia Quarteroni. Linguistic kernels for answer re-ranking in question answering systems. *Information Processing & Management*, 47(6): 825–842, November 2011. ISSN 03064573. doi: 10.1016/j.ipm.2010.06.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0306457310000518.

[128] Subhabrata Mukherjee and Pushpak Bhattacharyya. Sentiment Analysis in Twitter with Lightweight Discourse Analysis. In *COLING*, pages 1847–1864, 2012.

[129] Claudiu-Christian Musat, Alireza Ghasemi, and Boi Faltings. Sentiment analysis using a novel human computation game. In *Workshop on the People's Web Meets NLP, ACL*, 2012.

[130] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. Malt{P}arser: a language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[131] Jahna Otterbacher, Güne Erkan, and Dragomir R Radev. Using Random Walks for Question-focused Sentence Retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 915–922, 2005. doi: 10.3115/1220575.1220690.

[132] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010. ISSN 10414347. doi: 10.1109/TKDE.2009.191.

[133] Kishore Papineni, Salim Roukus, Todd Ward, and Wei-Jing Zhu. {BLEU}: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, 2002.

[134] Vincent Pascal, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[135] Slav Petrov and Ryan McDonald. Overview of the 2012 Shared Task on Parsing the Web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, 2012.

[136] Emily Pitler and Ani Nenkova. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 13–16, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1667583.1667589.

[137] Emily Pitler, Annie Louis, and Ani Nenkova. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 683–691, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-46-6. URL http://dl.acm.org/citation.cfm?id=1690219.1690241.

[138] Hoifung Poon and Pedro Domingos. Machine Reading: A" Killer App" for Statistical Relational AI. In *Statistical Relational Artificial Intelligence*, 2010.

[139] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. *In Proceedings of LREC*, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.7462.

[140] Rashmi Prasad, Aravind Joshi, and Bonnie Webber. Realization of discourse relations by other means: alternative lexicalizations. pages 1023–1031, August 2010. URL http://dl.acm.org/citation.cfm?id=1944566.1944684.

[141] Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. The biomedical discourse relation bank. *BMC Bioinformatics*, 12(1):188, 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-188. URL http://www.biomedcentral.com/1471-2105/12/188.

[142] Filip Radlinski, Paul N. Bennett, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance. *ACM SIGIR Forum*, 43(2):

46, December 2009. ISSN 01635840. doi: 10.1145/1670564.1670572. URL http://portal.acm.org/citation.cfm?doid=1670564.1670572.

[143] Balaji Polepalli Ramesh and Hong Yu. Identifying discourse connectives in biomedical text. In *AMIA Annual Symposium Proceedings*, volume 2010, page 657. American Medical Informatics Association, 2010.

[144] Rajesh Ranganath, Dan Jurafsky, and Dan McFarland. It's not you, it's me: detecting flirting and its misperception in speed-dates. In *NAACL*, 2009.

[145] Douglas Rohde, Laura Gonnerman, and David Plaut. An improved model of semantic similarity based on lexical co-occurrence. In *Cognitive Science*, 2009.

[146] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[147] Magnus Sahlgren. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, volume 5, 2005.

[148] Roy Schwartz. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *ACL*, 2011.

[149] Roy Schwartz, Omri Abend, and Ari Rappoport. Learnability-based syntactic annotation design. In *COLING*, 2012.

[150] Pnina Shachaf. The paradox of expertise: is the Wikipedia Reference Desk as good as your library? *Journal of Documentation*, 65(6):977–996, October 2009. ISSN 0022-0418. doi: 10.1108/00220410910998951. URL http://www.emeraldinsight.com/journals.htm?issn=0022-0418&volume=65&issue=6&articleid=1823656&show=html.

[151] Pnina Shachaf. Social reference: Toward a unifying theory. *Library & Information Science Research*, 32(1):66–76, 2010. URL http://www.sciencedirect.com/science/article/pii/S0740818809001406.

[152] Chirag Shah and Jefferey Pomerantz. Evaluating and predicting answer quality in community QA. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 411–418, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835518. URL http://doi.acm.org/10.1145/1835449.1835518.

[153] Amit Singhal, Chris Buckley, Mandar Mitra, and Ar Mitra. Pivoted Document Length Normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 1996. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.9950.

[154] Andrew Smith, Trevor Cohn, and Miles Osborne. Logarithmic opinion pools for conditional random fields. In *ACL*, 2005.

[155] Richard Socher, Eric Huan, Jeffrey Pennington, Andrew Ng, and Christopher Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS*, 2011.

[156] Lucia Specia, Sujay K. Jauhar, and Rada Mihalcea. SemEval-2012 Task 1: English Lexical Simplification. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, Montreal, Canada, 2012.

[157] Valentin I Spitkovsky, Hiyan Alshawi, Angel X Chang, and Daniel Jurafsky. Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1281–1290. Association for Computational Linguistics, 2011.

[158] Caroline Sporleder and Alex Lascarides. Using automatically labelled examples to classify rhetorical relations: an assessment. *Natural Language Engineering*, 14 (03):369–416, December 2006. ISSN 1351-3249. doi: 10.1017/S1351324906004451. URL http://dl.acm.org/citation.cfm?id=1394775.1394779.

[159] Qi Su, Chu-Ren Huang, and Helen Kai-yun Chen. Evidentiality for text trustworthiness detection. In *NLPLING '10 Proceedings of the 2010 Workshop on NLP and Linguistics:*

Finding the Common Ground, 2010. URL http://dl.acm.org/citation.cfm?id=1870168.

[160] Qian Sun, Rita Chattopadhyay, Sethuraman Panchanathan, and Jieping Ye. Two-stage weighting framework for multi-source domain adaptation. In *NIPS*, 2011.

[161] M Surdeanu, M Ciaramita, and H Zaragoza. Learning to rank answers on large online QA collections. *Proceedings of ACL-08: HLT*, pages 719–727, 2008.

[162] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online QA collections. In *In Proceedings of the 46th Annual Meeting for the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.164.2260.

[163] Maggy Anastasia Suryanto, Ee Peng Lim, Aixin Sun, and Roger H. L. Chiang. Quality-aware collaborative question answering. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining - WSDM '09*, page 142, New York, New York, USA, February 2009. ACM Press. ISBN 9781605583907. doi: 10.1145/1498759.1498820. URL http://dl.acm.org/citation.cfm?id=1498759.1498820.

[164] Charles Sutton, Michael Sindelar, and Andrew McCallum. Reducing weight undertraining in structured discriminative learning. In *NAACL*, 2006.

[165] Maite Taboada. Discourse markers as signals (or not) of rhetorical relations. *Journal of Pragmatics*, 38(4):567–592, April 2006. ISSN 03782166. doi: 10.1016/j.pragma.2005.09.010. URL http://dx.doi.org/10.1016/j.pragma.2005.09.010.

[166] Maite Taboada and William C Mann. Rhetorical Structure Theory: looking back and moving ahead. *Discourse Studies*, 8:423–459, 2006. doi: 10.1177/1461445606061881.

[167] Sali Tagliamonte. So who? Like how? Just what? Discourse markers in the conversations of Young Canadians. *Journal of Pragmatics*, 37:1896–1915, 2005.

[168] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. In *ICML*, 2011.

[169] YR Tausczik and JW Pennebaker. Predicting the perceived quality of online mathematics contributions from users' reputations. *CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011. URL http://dl.acm.org/citation.cfm?id=1979215.

[170] Gian Lorenzo Thione, Martin Van Den Berg, Livia Polanyi, and Chris Culy. Hybrid text summarization: Combining external relevance measures with structural analysis. In *Proceedings ACL Workshop Text Summarization Branches Out. Barcelona*, 2004.

[171] Sara Tonelli and Elena Cabrio. Hunting for Entailing Pairs in the Penn Discourse Treebank. In *Proceedings of COLING 2012*, pages 2653–2668, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-1162.

[172] Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. Cross-framework evaluation for statistical parsing. In *EACL*, 2012.

[173] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *ACL*, 2010.

[174] P. D Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010. URL http://www.jair.org/media/2934/live-2934-4846-jair.pdf.

[175] Miriam Urgelles-Coll. *The Syntax and Semantics of Discourse Markers*. Continuum, 2010. URL http://www.amazon.com/Semantics-Discourse-Theoretical-Linguistics-ebook/dp/B003LVZ48I.

[176] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2011.

[177] Erik Velldal, Lilja \Ovrelid, Jonathon Read, and Stephan Oepen. Speculation and Negation: Rules, Rankers, and

the Role of Synta. *Computational linguistics*, 38(2):369–410, 2012.

[178] Kiri Wagstaff. Machine learning that matters. In *ICML*, 2012.

[179] Xun Wang, Sujian Li, Jiwei Li, and Wenjie Li. Implicit Discourse Relation Recognition by Selecting Typical Training Examples. In *Proceedings of COLING 2012*, pages 2757–2772, Mumbai, 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-1168.

[180] Bonnie Webber and Aravind Joshi. Discourse Structure and Computation: Past, Present and Future. In *Association for Computational Linguistics*, page 42, 2012.

[181] Elizabeth G. Weber. *Varieties of Questions in English Conversation*. 1993. ISBN 902722613X. URL http://books.google.dk/books/about/Varieties_of_Questions_in_English_Conver.html?id=yycYdIciGP4C&pgis=1.

[182] Florian Wolf and Edward Gibson. Representing Discourse Coherence: A Corpus-Based Study. *Comput. Linguist.*, 31(2):249–288, 2005. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/0891201054223977. URL http://dx.doi.org/10.1162/0891201054223977.

[183] Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. Using a dependency parser to improve {SMT} for subject-object-verb languages. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT) 2009*, pages 245–253, Boulder, Colorado, 2009.

[184] Deniz Yuret, Laura Rimell, and Aydin Han. Parser evaluation using textual entailments. *Language Resources and Evaluation*, Published, 2012.

[185] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004.

[186] Zhi Min Zhou, Man Lan, Zheng Yu Niu, Yu Xu, and Jian Su. The effects of discourse connectives prediction on implicit discourse relation recognition. In *Proceedings of the 11th Annual Meeting of the Special Interest*

Group on Discourse and Dialogue, SIGDIAL '10, pages 139–146, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-85-5. URL http://dl.acm.org/citation.cfm?id=1944506.1944532.

[187] Zhi-Min Zhou, Man Lan, Zhen-Yu Niu, and Yue Lu. Exploiting user profile information for answer ranking in cQA. *WWW '12 Companion Proceedings of the 21st international conference companion on World Wide Web*, 2012. URL http://dl.acm.org/citation.cfm?id=2188199.