Unsupervised Entailment Detection between Dependency Graph Fragments

Marek Rei
Computer Laboratory
University of Cambridge
United Kingdom
Marek.Rei@cl.cam.ac.uk

Ted Briscoe
Computer Laboratory
University of Cambridge
United Kingdom
Ted.Briscoe@cl.cam.ac.uk

Abstract

Entailment detection systems are generally designed to work either on single words, relations or full sentences. We propose a new task – detecting entailment between dependency graph fragments of any type – which relaxes these restrictions and leads to much wider entailment discovery. An unsupervised framework is described that uses intrinsic similarity, multi-level extrinsic similarity and the detection of negation and hedged language to assign a confidence score to entailment relations between two fragments. The final system achieves 84.1% average precision on a data set of entailment examples from the biomedical domain.

1 Introduction

Understanding that two different texts are semantically similar has benefits for nearly all NLP tasks, including IR, IE, QA and Summarisation. Similarity detection is usually performed either on single words (synonymy) or full sentences and paragraphs (paraphrasing). A symmetric similarity relation implies that both elements can be interchanged (synonymy and paraphrasing), while directional similarity suggests that one fragment can be substituted for the other but not the opposite (hyponymy and entailment).

All of these language phenomena can be expressed using a single entailment relation. For paraphrases and synonyms the relation holds in both directions (*observe* $\leftrightarrow$ *notice*), whereas entailment and hyponymy are modelled as a unidirectional relation (*overexpress* $\rightarrow$ *express*). Such relations, however, can be defined between text fragments of any size and shape, ranging from a single word to a complete text segment. For example (*argues against* $\rightarrow$ *does not support*; *the protein has been implicated* $\leftrightarrow$ *the protein has been shown to be involved*).

We propose a new task – detecting entailment relations between any kinds of text fragments. A unified approach is not expected to perform better when compared to systems optimised only for a specific task (e.g. recognising entailment between sentences), but constructing a common theory to cover all text fragments has important benefits. A broader approach will allow for entailment discovery among a much wider range of fragment types for which no specialised systems exist. In addition, entailment relations can be found between different types of fragments (e.g. a predicate entailing an adjunct). Finally, a common system is much easier to develop and integrate with potential applications compared to having a separate system for each type of fragment.

In this paper we present a unified framework that can be used to detect entailment relations between fragments of various types and sizes. The system is designed to work with anything that can be represented as a dependency graph, including single words, constituents of various sizes, text adjuncts, predicates, relations and full sentences. The approach is completely unsupervised and requires only a large plain text corpus to gather information for calculating distributional similarity. This makes it ideal for the biomedical domain where the availability of annotated training data is limited. We apply these methods by using a background corpus

of biomedical papers and evaluate on a manually constructed dataset of entailing fragment pairs, extracted from biomedical texts.

2 Applications

Entailment detection between fragments is a vital step towards entailment generation – given text T, the system will have to generate all texts that either entail T or are entailed by T. This is motivated by applications in Relation Extraction, Information Retrieval and Information Extraction. For example, if we wish to find all genes that are synthesised in the larval tissue, the following IE query can be constructed (with *x* and *y* marking unknown variables):

(1) *x* is synthesised in the larval tissue

Known entailment relations can be used to modify the query: (*overexpression* $\rightarrow$ *synthesis*), (*larval fat body* $\rightarrow$ *larval tissue*) and (*the synthesis of x in y* $\leftrightarrow$ *x is synthesised in y*). Pattern (2) entails pattern (1) and would also return results matching the information need.

(2) the overexpression of *x* in the larval fat body

A system for entailment detection can automatically extract a database of entailing fragments from a large corpus and use them to modify any query given by the user. Recent studies have also investigated how complex sentence-level entailment relations can be broken down into smaller consecutive steps involving fragment-level entailment (Sammons et al., 2010; Bentivogli et al., 2010). For example:

(3) **Text**: The mitogenic effects of the B beta chain of fibrinogen are mediated through cell surface calreticulin.

Hypothesis: Fibrinogen beta chain interacts with CRP55.

To recognise that the hypothesis is entailed by the text, it can be decomposed into five separate steps involving text fragments:

1. *B beta chain of fibrinogen* $\rightarrow$ *Fibrinogen beta chain*
2. *calreticulin* $\rightarrow$ *CRP55*
3. *the mitogenic effects of x are mediated through y* $\rightarrow$ *y mediates the mitogenic effects of x*
4. *y mediates the mitogenic effects of x* $\rightarrow$ *y interacts with x*
5. *y interacts with x* $\rightarrow$ *x interacts with y*

This illustrates how entailment detection between various smaller fragments can be used to construct an entailment decision between more complicated sentences. However, only the presence of these constructions has been investigated and, to the best of our knowledge, no models currently exist for automated detection and composition of such entailment relations.

3 Modelling entailment between graph fragments

In order to cover a wide variety of language phenomena, a fragment is defined in the following way:

Definition 1. *A fragment is any connected subgraph of a directed dependency graph containing one or more words and the grammatical relations between them.*

This definition is intended to allow extraction of a wide variety of fragments from a dependency tree or graph representation of sentences found using any appropriate parser capable of returning such output (e.g. Kübler et al., 2009). The definition covers single- or multi-word constituents functioning as dependents (e.g. *sites*, *putative binding sites*), text adjuncts (*in the cell wall*), single- or multi-word predicates (* *binds to receptors in the airways*) and relations (* *binds and activates* *) including ones with 'internal' dependent slots (* *inhibits* * *at* *), some of which may be fixed in the fragment (* *induces autophosphorylation of* * *in* * *cells*), and also full sentences[1]. An example dependency graph and some selected fragments can be seen in Figure 1.

Our aim is to detect semantically similar fragments which can be substituted for each other in text, resulting in more general or more specific versions of the same proposition. This kind of similarity can be thought of as an entailment relation and we define entailment between two fragments as follows:

[1]The asterisks (*) are used to indicate missing dependents in order to increase the readability of the fragment when represented textually. The actual fragments are kept in graph form and have no need for them.

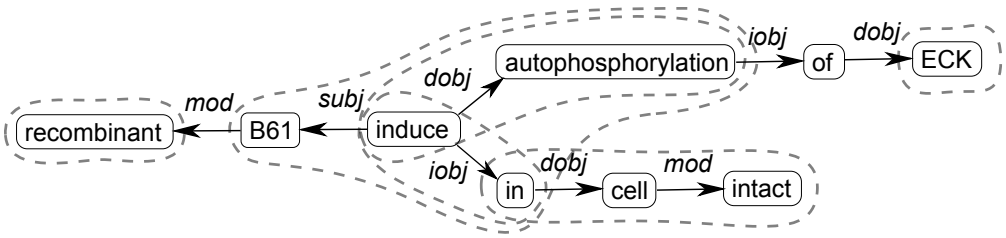

Figure 1: Dependency graph for the sentence: *Recombinant B61 induces autophosphorylation of ECK in intact cells*. Some interesting fragments are marked by dotted lines.

Definition 2. *Fragment A entails fragment B (A → B) if A can be replaced by B in sentence S and the resulting sentence S' can be entailed from the original one (S → S').*

This also requires estimating entailment relations between sentences, for which we use the definition established by Bar-Haim et al. (2006):

Definition 3. *Text T entails hypothesis H (T → H) if, typically, a human reading T would infer that H is most likely true.*

We model the semantic similarity of fragments as a combination of two separate directional similarity scores:

1. **Intrinsic similarity**: how similar are the components of the fragments.
2. **Extrinsic similarity**: how similar are the contexts of the fragments.

To find the overall score, these two similarity measures are combined linearly using a weighting parameter α:

$$Score(A \rightarrow B) = \alpha \times IntSim(A \rightarrow B) + (1 - \alpha) \times ExtSim(A \rightarrow B)$$

In this paper $f(A \rightarrow B)$ designates an asymmetric function between A and B. When referring only to single words, lowercase letters (a,b) are used; when referring to fragments of any size, including single words, then uppercase letters are used (A,B).

$Score(A \rightarrow B)$ is the confidence score that fragment A entails fragment B – higher score indicates higher confidence and 0 means no entailment. $IntSim(A \rightarrow B)$ is the intrinsic similarity between two fragments. It can be any function that compares them, for example by matching the structure of one fragment to another, and outputs a similarity score in the range of $[0, 1]$. $ExtSim(A \rightarrow B)$ is a measure of extrinsic similarity that compares the contexts of the two fragments. α is set to 0.5 for an unsupervised approach but the effects of tuning this parameter are further analysed in Section 5.

The directional similarity score is first found between words in each fragment, which are then used to calculate the score between the two fragments.

3.1 Intrinsic similarity

$IntSim(A \rightarrow B)$ is the intrinsic similarity between the two words or fragments. In order to best capture entailment, the measure should be non-symmetrical. We use the following simple formula for word-level score calculation:

$$IntSim(a \rightarrow b) = \frac{length(c)}{length(b)}$$

where c is the longest common substring for a and b. This measure will show the ratio of b that is also contained in a. For example:

$$IntSim(\textit{overexpress} \rightarrow \textit{expression}) = 0.70$$

$$IntSim(\textit{expression} \rightarrow \textit{overexpress}) = 0.64$$

The intrinsic similarity function for fragments is defined using an injective function between components of A and components of B:

$$IntSim(A \rightarrow B) = \frac{Mapping(A \rightarrow B)}{|B|}$$

where $Mapping(A \rightarrow B)$ is a function that goes through all the possible word pairs $\{(a, b)|a \in A, b \in B\}$ and at each iteration connects the one

with the highest entailment score, returning the sum of those scores. Figure 2 contains pseudocode for the mapping process. Dividing the value of $Mapping(A \rightarrow B)$ by the number of components in B gives an asymmetric score that indicates how well B is covered by A. It returns a lower score if B contains more elements than A as some words cannot be matched to anything. While there are exceptions, it is common that if B is larger than A, then it cannot be entailed by A as it contains more information.

while $unused\ elements\ in\ A\ and\ B$ **do**
  $bestScore = 0$
  for $a \in A, b \in B, a\ and\ b\ are\ unused$ **do**
    if $Score(a \rightarrow b) > bestScore$ **then**
      $bestScore = Score(a \rightarrow b)$
    end if
  end for
  $total+ = bestScore$
end while
return $total$

Figure 2: Pseudocode for mapping between two fragments

The word-level entailment score $Score(a \rightarrow b)$ is directly used to estimate the entailment score between fragments, $Score(A \rightarrow B)$. In this case we are working with two levels – fragments which in turn consist of words. However, this can be extended to a truly recursive method where fragments consist of smaller fragments.

3.2 Extrinsic similarity

The extrinsic similarity between two fragments or words is modelled using measures of directional distributional similarity. We define a context relation as a tuple (a, d, r, a') where a is the main word, a' is a word connected to it through a dependency relation, r is the label of that relation and d shows the direction of the relation. The tuple $f : (d, r, a')$ is referred to as a feature of a.

To calculate the distributional similarity between two fragments, we adopt an approach similar to Weeds et al. (2005). Using the previous notation, (d, r, a') is a feature of fragment A if (d, r, a') is a feature for a word which is contained in A. The general algorithm for feature collection is as follows:

1. Find the next instance of a fragment in the background corpus.
2. For each word in the fragment, find dependency relations which connect to words not contained in the fragment.
3. Count these dependency relations as distributional features for the fragment.

For example, in Figure 1 the fragment *(* induces * in *)* has three features: *(1, subj, B61)*, *(1, dobj, autophosphorylation)* and *(1, dobj, cell)*.

The BioMed Central[2] corpus of full papers was used to collect distributional similarity features for each fragment. 1000 papers were randomly selected and separated for constructing the test set, leaving 70821 biomedical full papers. These were tokenised and parsed using the RASP system (Briscoe et al., 2006) in order to extract dependency relations.

We experimented with various schemes for feature weighting and found the best one to be a variation of Dice's coefficient (Dice, 1945), described by Curran (2003):

$$w_A(f) = \frac{2P(A, f)}{P(A, *) + P(*, f)}$$

where $w_A(f)$ is the weight of feature f for fragment A, $P(*, f)$ is the probability of the feature appearing in the corpus with any fragment, $P(A, *)$ is the probability of the fragment appearing with any feature, and $P(A, f)$ is the probability of the fragment and the feature appearing together.

Different measures of distributional similarity, both symmetrical and directonal, were also tested and $ClarkeDE$ (Clarke, 2009) was used for the final system as it achieved the highest performance on graph fragments:

$$ClarkeDE(A \rightarrow B) = \frac{\sum_{f \in F_A \cap F_B} min(w_A(f), w_B(f))}{\sum_{f \in F_A} w_A(f)}$$

where F_A is the set of features for fragment A and $w_A(f)$ is the weight of feature f for fragment A. It quantifies the weighted coverage of the features of A by the features of B by finding the sum of minimum weights.

[2] http://www.biomedcentral.com/info/about/datamining/

The $ClarkeDE$ similarity measure is designed to detect whether the features of A are a proper subset of the features of B. This works well for finding more general versions of fragments, but not when comparing fragments which are roughly equal paraphrases. As a solution we constructed a new measure based on the symmetrical Lin measure (Lin, 1998).

$$LinD(A \rightarrow B) = \frac{\sum_{f \in F_A \cap F_B} [w_A(f) + w_B(f)]}{\sum_{f \in F_A} w_A(f) + \sum_{f \in F_A \cap F_B} w_B(f)}$$

Compared to the original, the features of B which are not found in A are excluded from the score calculation, making the score non-symmetrical but more balanced compared to $ClarkeDE$. We applied this for word-level distributional similarity and achieved better results than with other common similarity measures.

The LinD similarity is also calculated between fragment levels to help detect possible paraphrases. If the similarity is very high (greater than 85%), then a symmetric relationship between the fragments is assumed and the value of $LinD$ is used as $ExtSim$. Otherwise, the system reverts to the $ClarkeDE$ measure for handling unidirectional relations.

3.3 Hedging and negation

Language constructions such as hedging and negation typically invert the normal direction of an entailment relation. For example, (*biological discovery* $\rightarrow$ *discovery*) becomes (*no biological discovery* $\leftarrow$ *no discovery*) and (*is repressed by* $\rightarrow$ *is affected by*) becomes (*may be repressed by* $\leftarrow$ *is affected by*).

Such cases are handled by inverting the direction of the score calculation if a fragment is found to contain a special cue word that commonly indicates hedged language or negation. In order to find the list of indicative hedge cues, we analysed the training corpus of CoNLL 2010 Shared Task (Farkas et al., 2010) which is annotated for speculation cues and scopes. Any cues that occurred less than 5 times or occurred more often as normal text than as cue words were filtered out, resulting in the following list:

(4) *suggest, may, might, indicate that, appear, likely, could, possible, whether, would, think, seem, probably, assume, putative, unclear, propose, imply, possibly*

For negation cues we used the list collected by Morante (2009):

(5) *absence, absent, cannot, could not, either, except, exclude, fail, failure, favor over, impossible, instead of, lack, loss, miss, negative, neither, nor, never, no, no longer, none, not, rather than, rule out, unable, with the exception of, without*

This is a fast and basic method for estimating the presence of hedging and negation in a fragment. When dealing with longer texts, the exact scope of the cue word should be detected, but for relatively short fragments the presence of a keyword acts as a good indication of hedging and negation.

4 Evaluation

A "pilot" dataset was created to evaluate different entailment detection methods between fragments[3]. In order to look for valid entailment examples, 1000 biomedical papers from the BioMed Central full-text corpus were randomly chosen and analysed. We hypothesised that two very similar sentences originating from the same paper are likely to be more and less general versions of the same proposition. First, the similarities between all sentences in a single paper were calculated using a bag-of-words approach. Then, ten of the most similar but non-identical sentence pairs from each paper were presented for manual review and 150 fragment pairs were created based on these sentences, 100 of which were selected for the final set.

When applied to sentence-level entailment extraction, similar methods can suffer from high lexical overlap as sentences need to contain many matching words to pass the initial filter. However, for the extraction of entailing fragments most of the matching words are discarded and only the non-identical fragments are stored, greatly reducing the overlap problem. Experiments in Section 5 demonstrate that a simple bag-of-words approach performs rather poorly on the task, confirming that the extraction method produces a diverse selection of fragments.

[3] http://www.cl.cam.ac.uk/~mr472/entailment/

Two annotators assigned a relation type to candidate pairs based on how well one fragment can be substituted for the other in text while preserving meaning ($A \leftrightarrow B$, $A \rightarrow B$, $A \leftarrow B$ or $A \neq B$). Cohen's Kappa between the annotators was 0.88, indicating very high agreement. Instances with disagreement were then reviewed and replaced for the final dataset.

Each fragment pair has two binary entailment decisions (one in either direction) and the set is evenly balanced, containing 100 entailment and 100 non-entailment relations. An example sentence with the first fragment is also included. Fragment sizes range from 1 to 20 words, with the average of 2.86 words.

The system assigns a score to each entailment relation, with higher values indicating higher confidence in entailment. All the relations are ranked based on their score, and average precision (AP) is used to evaluate the performance:

$$\text{AP} = \frac{1}{R}\sum_{i=1}^{N}\frac{E(i) \times CorrectUpTo(i)}{i}$$

where R is the number of correct entailment relations, N is the number of possible relations in the test set, $E(i)$ is 1 if the i-th relation is entailment in the gold standard and 0 otherwise, and $CorrectUpTo(i)$ is the number of correctly returned entailment relations up to rank i. Average precision assigns a higher score to systems which rank correct entailment examples higher in the list.

As a secondary measure we also report the Break-Even Point (BEP) which is defined as precision at the rank where precision is equal to recall. Using the previous annotation, this can also be calculated as precision at rank R:

$$BEP = \frac{CorrectUpTo(R)}{R}$$

BEP is a much more strict measure, treating the task as binary classification and ignoring changes to the ranks within the classes.

5 Results

The test set is balanced, therefore random guessing would be expected to achieve an AP and BEP of 0.5 which can be regarded as the simplest (random) baseline. Table 1 contains results for two more basic approaches to the task. For the bag-of-words (BOW) system, the score of A entailing B is the proportion of words in B that are also contained in A.

$$Score_{bow}(A \rightarrow B) = \frac{|\{b|b \in A, B\}|}{|\{b|b \in B\}|}$$

We also tested entailment detection when using only the directional distributional similarity between fragments as it is commonly done for words. While both of the systems perform better than random, the results are much lower than those for more informed methods. This indicates that even though there is some lexical overlap between the fragments, it is not enough to make good decisions about the entailment relations.

System type	**AP**	**BEP**
Random baseline	0.500	0.500
BOW	0.657	0.610
Distributional similarity	0.645	0.480

Table 1: Results for basic approaches.

Table 2 contains the results for the system described in Section 3. We start with the most basic version and gradually add components. Using only the intrinsic similarity, the system performs better than any of the basic approaches, delivering 0.71 AP.

System type	**AP**	**BEP**
Intrinsic similarity only	0.710	0.680
+ Word ExtSim	0.754	0.710
+ Fragment ExtSim	0.801	0.710
+ Negation & hedging	0.831	0.720
+ Paraphrase check	0.841	0.720

Table 2: Results for the system described in Section 3. Components are added incrementally.

Next, the extrinsic similarity between words is included, raising the AP to 0.754. When the string-level similarity fails, the added directional distributional similarity helps in mapping semantically equivalent words to each other.

The inclusion of extrinsic similarity between fragments gives a further increase, with AP of 0.801. The 4.5% increase shows that while fragments are

larger and occur less often in a corpus, their distributional similarity can still be used as a valuable component to detect semantic similarity and entailment.

Checking for negation and hedge cues raises the AP to 0.831. The performance is already high and a 3% improvement shows that hedging and negation affect fragment-level entailment and other components do not manage to successfully capture this information.

Finally, applying the fragment-level check for paraphrases with a more appropriate distributional similarity measure, as described in Section 3.2, returns an AP of 0.841. The results of this final configuration are significantly different compared to the initial system using only intrinsic similarity, according to the Wilcoxon signed rank test at the level of 0.05.

The formula in Section 3 contains parameter α which can be tuned to adjust the balance of intrinsic and extrinsic similarity. This can be done heuristically or through machine learning methods and different values can be used for fragments and words. In order to investigate the effects of tuning on the system, we tried all possible combinations of α_{word} and $\alpha_{fragment}$ with values between 0 and 1 at increments of 0.05. Table 3 contains results for some of these experiments.

α_{word}	$\alpha_{fragment}$	AP	BEP
0.5	0.5	0.841	0.720
*	0.0	0.656	0.480
0.0	1.0	0.813	0.720
1.0	1.0	0.765	0.690
0.45	0.65	0.847	0.740

Table 3: Results of tuning the weights for intrinsic and distributional similarity.

The best results were obtained with $\alpha_{word} = 0.45$ and $\alpha_{fragment} = 0.65$, resulting in 0.847 AP and 0.74 BEP. This shows that parameter tuning can improve the results, but the 0.6% increase is modest and a completely unsupervised approach can give competitive results. In addition, the optimal values of α are close to 0.5, indicating that all four components (intrinsic and distributional similarities between words and fragments) are all contributing to the performance of the final system.

6 Previous work

Most work on entailment has focused on comparing sentences or paragraphs. For example, Haghighi et al. (2005) represent sentences as directed dependency graphs and use graph matching to measure semantic overlap. This method also compares the dependencies when calculating similarity, which supports incorporation of extra syntactic information. Hickl et al. (2006) combine lexico-syntactic features and automatically acquired paraphrases to classify entailing sentences. Lintean and Rus (2009) make use of weighted dependencies and word semantics to detect paraphrases. In addition to similarity they look at dissimilarity between two sentences and use their ratio as the confidence score for paraphrasing.

Lin and Pantel (2001) were one of the first to extend the distributional hypothesis to dependency paths to detect entailment between relations. Szpektor et al. (2004) describe the TEASE method for extracting entailing relation templates from the Web. Szpektor and Dagan (2008) use the distributional similarity of arguments to detect unary template entailment, whilst Berant et al. (2010) apply it to binary relations in focused entailment graphs.

Snow et al. (2005) described a basic method of syntactic pattern matching to automatically discover word-level hypernym relations from text. The use of directional distributional similarity measures to find inference relations between single words is explored by Kotlerman et al. (2010). They propose new measures based on feature ranks and compare them with existing ones for the tasks of lexical expansion and text categorisation.

In contrast to current work, each of the approaches described above only focuses on detecting entailment between specific subtypes of fragments (either sentences, relations or words) and optimising the system for a single scenario. This means only limited types of entailment relations are found and they cannot be used for entailment generation or compositional entailment detection as described in Section 2.

MacCartney and Manning (2008) approach sentence-level entailment detection by breaking the problem into a sequence of atomic edits linking the premise to the hypothesis. Entailment relations are then predicted for each edit, propagated up through

a syntax tree and then used to compose the resulting entailment decision. However, their system focuses more on natural logic and uses a predefined set of compositional rules to capture a subset of valid inferences with high precision but low recall. It also relies on a supervised classifier and information from WordNet to reach the final entailment decision.

Androutsopoulos and Malakasiotis (2010) have assembled a survey of different tasks and approaches related to paraphrasing and entailment. They describe three different goals (paraphrase recognition, generation and extraction) and analyse various methods for solving them.

7 Conclusion

Entailment detection systems are generally developed to work on specific text units – either single words, relations, or full sentences. While this reduces the complexity of the problem, it can also lead to important information being disregarded. In this paper we proposed a new task – detecting entailment relations between any kind of dependency graph fragments. The definition of a fragment covers the language structures mentioned above and also extends to others that have not been fully investigated in the context of entailment recognition (such as multi-word constituents, predicates and adjuncts).

To perform entailment detection between various types of dependency graph fragments, a new system was built that combines the directional intrinsic and extrinsic similarities of two fragments to reach a final score. Fragments which contain hedging or negation are identified and their score calculation is inverted to better model the effect on entailment. The extrinsic similarity is found with two different distributional similarity measures, first checking for symmetric similarity and then for directional containment of distributional features. The system was evaluated on a manually constructed dataset of fragment-level entailment relations from the biomedical domain and each of the added methods improved the results.

Traditionally, the method for entailment recognition is chosen based on what appears optimal for the task – either structure matching or distributional similarity. Our experiments show that the combination of both gives the best performance for all fragment types. It is to be expected that single words will benefit more from distributional measures while full sentences get matched by their components. However, this separation is not strict and evidence from both methods can be used to strengthen the decision.

The experiments confirmed that entailment between dependency graph fragments of various types can be detected in a completely unsupervised setting, without the need for specific resources or annotated training data. As our method can be applied equally to any domain and requires only a large plain text corpus, we believe it is a promising direction for research in entailment detection. This can lead to useful applications in biomedical information extraction where manually annotated datasets are in short supply.

We have shown that a unified approach can be used to detect entailment relations between dependency graph fragments. This allows for entailment discovery among a wide range of fragment types, including ones for which no specialised systems currently exist. The framework for fragment-level entailment detection can be integrated into various applications that require identifying and rewriting semantically equivalent phrases - for example, query expansion in IE and IR, text mining, sentence-level entailment composition, relation extraction and protein-protein interaction extraction.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38(7):135–187.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–9. Citeseer.

Luisa Bentivogli, Elena Cabrio, Ido Dagan, Danilo Giampiccolo, Medea Lo Leggio, and Bernardo Magnini. 2010. Building Textual Entailment Specialized Data Sets: a Methodology for Isolating Linguistic Phenomena Relevant to Inference. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, number Section 6, pages 1220–1229. Association for Computational Linguistics.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, number July, pages 77–80, Sydney, Australia. Association for Computational Linguistics.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, number March, pages 112–119. Association for Computational Linguistics.

James Richard Curran. 2003. *From distributional to semantic similarity*. Ph.D. thesis, University of Edinburgh.

Lee R. Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Richárd Farkas, Veronika Vincze, György Móra, Jànos Csirik, and György Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 1–12. Association for Computational Linguistics.

Aria D. Haghighi, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via graph matching. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA. Association for Computational Linguistics.

Andrew Hickl, Jeremy Bensley, John Williams, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing textual entailment with LCC's GROUNDHOG system. In *Proceedings of the Second PASCAL Challenges Workshop*.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(04):359–389.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency Parsing. *Synthesis Lectures on Human Language Technologies*, 2:1–127.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(04):343–360.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics.

Mihain C. Lintean and Vasile Rus. 2009. Paraphrase Identification Using Weighted Dependencies and Word Semantics. In *Proceedings of the FLAIRS-22*, volume 22, pages 19–28.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 521–528. Association for Computational Linguistics.

Roser Morante. 2009. Descriptive analysis of negation cues in biomedical texts. In *Proceedings of the Seventh International Language Resources and Evaluation (LREC10)*, pages 1429–1436.

Mark Sammons, V.G. Vinod Vydiswaran, and Dan Roth. 2010. Ask not what textual entailment can do for you... In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 1199–1208. Association for Computational Linguistics.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics - COLING '08*, pages 849–856, Morristown, NJ, USA. Association for Computational Linguistics.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP*, volume 4, pages 41–48.

Julie Weeds, David Weir, and Bill Keller. 2005. The distributional similarity of sub-parses. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 7–12, Morristown, NJ, USA. Association for Computational Linguistics.