Survey in Textual Entailment

Swapnil Ghuge
Arindam Bhattacharya

1 Introduction

Variability of semantic expression is a fundamental phenomenon of a natural language where same meaning can be expressed by different texts. Natural Language Processing applications like Question Answering, Summarization, Information Retrieval systems *etc.* often demand a generic framework to capture major semantic inferences in order to deal with the challenges created by this phenomenon. Textual Entailment can provide such framework.

Textual Entailment can be defined as the phenomenon of inferring a text from another. Entailment is a directional relation between two texts. This relation holds when the truth of one text fragment follows from the truth of the other. Conventionally, the entailing fragment is called as *text* and entailed one is called as *hypothesis*. Textual entailment is classically defined as:

Classical Definition: A text t entails hypothesis h if h is true in every circumstance of possible world in which t is true.

This definition is very strict since it requires truthfulness of h in all the instances where t is true. Due to uncertainties in the real world applications, this definition is not very helpful. Hence applied definition of Textual Entailment is presented:

Applied Definition: A text t entails hypothesis h if human reading h will infer that h is **most likely** true.

Again, this definition is abstract for systems trying to implement Textual Entailment. Thus mathematically precise and computable definition using probabilities is provided:

Mathematical Definition (Glickman et al., 2005): Hypothesis h is entailed by text t if

$$P(h \textit{ is true} \mid t) > P(h \textit{ is true}) \tag{1}$$

$P(h \textit{ is true} \mid t)$ is the Entailment Confidence and can be considered as a measure of surety of entailment.

A better insight can be obtained from the following examples:

1. **T:** *iTunes software has seen strong sales in Europe.*
   H: *Strong sales of iTunes in Europe.*
   (RTE-1 Development Dataset, ID 13)

2. **T:** *Cavern Club sessions paid the Beatles 15 evenings and 5 lunchtime.*
   H: *The Beatles perform at Cavern Club at lunchtime.*

3. **T:** *American Airlines began laying off hundreds of flight attendants on Tuesday.*
   H: *European Airlines laid off their flight attendants.*

4. **T:** *Oracle had fought to keep the forms from being released.*
   H: *Oracle released a confidential document.*
   (RTE-1 Development Dataset, ID 12)

According to the classical definition of entailment, only (1) is a true entailment. While (1) and (2) both are valid entailments according to the applied definition, truthfulness of *hypothesis* in (3) cannot be determined by truthfulness of the *text*. A clear contradiction can be seen in (4). From the mathematical definition, it can be deduced that entailment confidence (given as $P(h\ is\ true \mid t)$) will be very high for (1). Confidence will be high for (2) though it will not be as high as that of (1). (3) will yield a lower score since entailment can not be shown. In case of (4), a clear contradiction states that whenever t is true, h has to be false and thus confidence score in this case is 0.

In the subsequent sections, a variety of approaches used to recognize text entailment will be discussed.

2 Evaluation Forum: RTE Challenges

The goal of the RTE competition is (as stated in PASCAL RTE challenge):

> The recognizing textual entailment is an attempt to promote an abstract generic task that captures major semantic inference needs across applications.

This encourages the creation of the generic framework to capture semantic inference.

2.1 Evaluation measures

The results of the RTE tasks are evaluated against human gold standard. The following are the metrics used:

- **Accuracy** The accuracy of a Text entailment system is the ratio of correct entailment decision to the total number of entailment problem.
- **Precision** The precision of Text entailment system is the ratio of number of correctly predicted entailment to the number of predicted entailment.

- **Recall** The recall of Text entailment system is the ratio of number of correctly predicted entailment to the actual number of correct entailments.

2.2 RTE 1 (2005)

The first PASCAL Recognizing Textual Entailment Challenge (15 June 2004 - 10 April 2005) (Dagan et al., 2005) provided the first benchmark for the entailment task and it was an initial attempt to form a generic empirical task that captures major semantic inferences across applications. The challenge raised noticeable attention in the research community, attracting 17 submissions from research groups worldwide. The relatively low accuracy achieved by the participating systems suggests that the entailment task is indeed a challenging one, with a wide room for improvement.

Participants in the evaluation exercise were provided with pairs of small text snippets (one or more sentences in English), which are termed Text-Hypothesis (T-H) pairs. The data set includes over 1000 English $T-H$ pairs from the news domain (political, economical, *etc.*). Examples are manually tagged for entailment (*i.e.* whether T entails H or not) by human annotators and are divided into a Development Set (one third of the data) and a Test Set (two thirds of the data). The dataset was collected with respect to different text processing applications, such as Question Answering, Information Extraction, Information Retrieval, Multi-document Summarization, Paraphrase Acquisition, *etc.* Examples showed different levels of entailment reasoning such as lexical, syntactic, morphological and logical. Participating systems had to decide for each $T-H$ pair whether T indeed entails H or not, and results were compared to the manual gold standard.

An interesting observation from the results was that the performance of the system did not correlate with the system complexity. The maximum precision obtained was 0.7, by Perez et al., using simple word overlap techniques.

2.3 RTE 2 (2006)

Similar to the first RTE challenge, the main task is judging whether a hypothesis H is entailed by a text T. One of the main goals for the RTE-2 data set is to provide more *realistic* text-hypothesis examples, based mostly on outputs of actual systems (Bar-Haim et al., 2006). RTE 2 received 23 submissions which presented diverse approaches and research direction. The best results obtained were considerably higher than RTE 1's state of the art.

Focus of the dataset was on the four application settings: Question Answering (QA), Information Retrieval (IR), Information Extraction (IE) and Multi-document Summarization. Each portion of the data set includes typical $T-H$ examples that correspond to success and failure cases of such applications. The highest accuracy achieved was 0.7538 with a precision of 0.8082 by Andrew Hickl. Machine Learning Classification-based approach was used.

2.4 RTE 3 (2007)

RTE 3 follows the same basic structure of the previous challenges, in order to facilitate the participation of newcomers and to allow "veterans" to assess the improvements of their systems in a comparable test exercise (Giampiccolo et al., 2007b). Nevertheless, the following innovations are introduced to make the challenge more stimulating and, at the same time, to encourage collaboration between system developers:

- a limited number of longer texts, *i.e.* up to a paragraph - in order to move toward more comprehensive scenarios which incorporate the need for discourse analysis. However, the majority of examples were kept similar to those in the previous challenges, providing pairs with relatively short texts.
- provision of an RTE Resource Pool where contributors have the possibility to share the resources they use.
- a pilot task, "Extending the Evaluation of Inferences from Texts", set up by US National Institute of Standards and Technology (NIST), which explored two other tasks closely related to textual entailment: differentiating unknown from false/contradicts and providing justifications for answers.

The best result was obtained by Hickl and Bensley (Hickl and Bensley, 2007) with accuracy of 80% and precision of 88.15%. It was a considerable amount of improvement over the existing state of art. Approach was to extract all possible discourse commitments (publicly-held beliefs) from the text and match them with hypothesis.

2.5 RTE 4 (2008)

In 2008 the Recognizing Textual Entailment challenge (RTE-4) was proposed for the first time as a track at the Text Analysis Conference (TAC) (Giampiccolo et al., 2007a). RTE-4 included the 3-way classification task that was piloted in RTE-3. The goal of making a three-way decision of *Entailment*, *Contradiction* and *Unknown* is to drive systems to make more precise informational distinctions; a hypothesis being unknown on the basis of a text should be distinguished from a hypothesis being shown false/contradicted by a text. The three way classification task lead to a slight decrease of the accuracies of the system. The highest accuracy was 0.746 again by LCC's Bensley and Hickl using the LCC's GROUNDHOG system (Hickl et al., 2006).

2.6 RTE 5 (2009)

RTE-5 was proposed as a track at Text Analysis Conference in 2009. The main RTE-5 task was similar to the RTE-4 task, with the following changes(Bentivogli et al., 2009):

- The average length of the Texts was higher.

- Texts came from a variety of sources and were not edited from their source documents. Thus, systems were asked to handle real text that may include typographical errors and ungrammatical sentences.
- A development set was released.
- The textual entailment recognition task was based on only three application settings: QA, IE, and IR.
- Ablation tests were made mandatory.

In addition to the main task (Textual Entailment Recognition), a new Textual Entailment Search pilot task was offered that was situated in the summarization application setting, where the task was to find all Texts in a set of documents that entail a given Hypothesis. 21 teams participated in the RTE-5 challenge out of which 20 submitted the systems for main task while 8 teams tried to tackle the pilot task.

In the main task, the highest accuracy obtained was 0.6833. In the search pilot task, the highest F-score obtained was 45.59.

Search pilot task introduced the real interaction between RTE task and Summarization task allowing the analysis of the impact of textual entailment recognition on a real NLP application.

2.7 RTE 6 (2010)

The RTE-6 tasks focus on recognizing textual entailment in two application settings: Summarization and Knowledge Base Population.

- **Main Task (Summarization scenario)**: Given a corpus and a set of "candidate" sentences retrieved by Lucene from that corpus, RTE systems are required to identify all the sentences from among the candidate sentences that entail a given Hypothesis. The RTE-6 Main Task is based on the TAC Update Summarization Task. In the Update Summarization Task, each topic contains two sets of documents ("A" and "B"), where all the "A" documents chronologically precede all the "B" documents. An RTE-6 Main Task "corpus" consists of 10 "A" documents, while Hypotheses are taken from sentences in the "B" documents.
- **KBP Validation Pilot (Knowledge Base Population scenario)**: Based on the TAC Knowledge Base Population (KBP) Slot-Filling task, the new KBP validation pilot task is to determine whether a given relation (Hypothesis) is supported in an associated document (Text). Each slot fill that is proposed by a system for the KBP Slot-Filling task would create one evaluation item for the RTE-KBP Validation Pilot: The Hypothesis would be a simple sentence created from the slot fill, while the Text would be the source document that was cited as supporting the slot fill

Thus RTE-6 did not include the traditional RTE Main Task of judging the entailment between a pair of isolated Text-Hypothesis pair. The Main Task was

based only on the Summarization application setting and was similar to the pilot Search Task introduced in RTE-5 with following changes:

- RTE-6 hypotheses were taken from sentences in the "B" documents, rather than from human-authored summaries of the "A" documents.
- A smaller number of candidate sentences were retrieved by Lucene baseline instead of searching for entailing sentences from the entire corpus.
- The exploratory effort on resource evaluation continued through ablation tests for the new RTE-6 Main Task.

The change in the task setting increased the difficulty level of the task which was reflected in the results. The highest F-measure was 0.4801. Debarghya from IIT, Bomabay, was placed third with the F-score of 0.4756 (Bhattacharya, 2012). The base-line F-score was 34.63.

2.8 RTE 7 (2011)

The RTE-7 tasks focus on recognizing textual entailment in two application settings: *Summarization and Knowledge Base Population*.

- **Main Task (Summarization setting):** Given a corpus and a set of "candidate" sentences retrieved by Lucene from that corpus, RTE systems are required to identify all the sentences from among the candidate sentences that entail a given Hypothesis. The RTE-7 Main Task is based on the TAC Update Summarization Task. In the Update Summarization Task, each topic contains two sets of documents ("A" and "B"), where all the "A" documents chronologically precede all the "B" documents. An RTE-7 Main Task "corpus" consists of 10 "A" documents, while Hypotheses are taken from sentences in the "B" documents.
- **Novelty Detection Sub-task (Summarization setting):** In the Novelty Detection variant of the Main Task, systems are required to judge if the information contained in each H (based on text snippets from B summaries) is novel with respect to the information contained in the A documents related to the same topic. If entailing sentences are found for a given H, it means that the content of H is not new; if no entailing sentences are detected, it means that information contained in the H is novel.
- **KBP Validation Task (Knowledge Base Population setting):** Based on the TAC Knowledge Base Population (KBP) Slot-Filling task, the KBP validation task is to determine whether a given relation (Hypothesis) is supported in an associated document (Text). Each slot fill that is proposed by a system for the KBP Slot-Filling task would create one evaluation item for the RTE-KBP Validation Task: The Hypothesis would be a simple sentence created from the slot fill, while the Text would be the source document that was cited as supporting the slot fill.

A total of thirteen teams participated in this competition. The best F-score was 0.4200. Arindam, from IIT, Bombay scored 0.3587.

RTE	T-H Pairs	Notable Feature
1	287	Mostly lexical systems. Results do not correlate with system complexities.
2	800	Application based focus. Mainly on Question Answering (QA).
3	800	Longer sentences. RTE resource pool was created along with it.
4	1000	Introduction of 3 way tasks.
5	600	Unedited real world text. Tasks based on QA and Information Retrieval (IR).
6	15,955	221 hypothesis with upto 100 Lucene retireved candidates.
7	21,420	284 hypothesis with Lucene retrieved candidates. Text upto paragraph long. Based on summarization setting.

Table 1: Evolution of RTE Challenges

The increasing complexity of the tasks and the data-set in RTE challenges is reflected in the results. Lower accuracies show that there is still wide room for improvement in the field of Textual Entailment. The figures are summarized below.

- Highest accuracy score was achieved in RTE-3. 3-way classification in RTE-4 and longer texts in RTE-5 resulted in lower accuracies.
- Most of the systems got accuracies between 55% to 65% which shows that task of RTE is very challenging.
- Participation increased every year. Diverse approaches and research directions have been presented which started to fulfill the purpose of the RTE challenges.

3 Approaches to Recognize Text Entailment

Diverse nature and a wide number of entailment triggers increase the difficulty of Textual Entailment task to a great extent. From morphological similarity to lexical overlapping and from shallow syntactic comparison to deep semantics extraction - different approaches are required to deal with different types of entailment triggers. After the introduction of RTE challenges in 2005, a spectrum of approaches have been proposed every year. A common approach is to re-represent both the *text* and *hypothesis* (figure 4) and determine if the

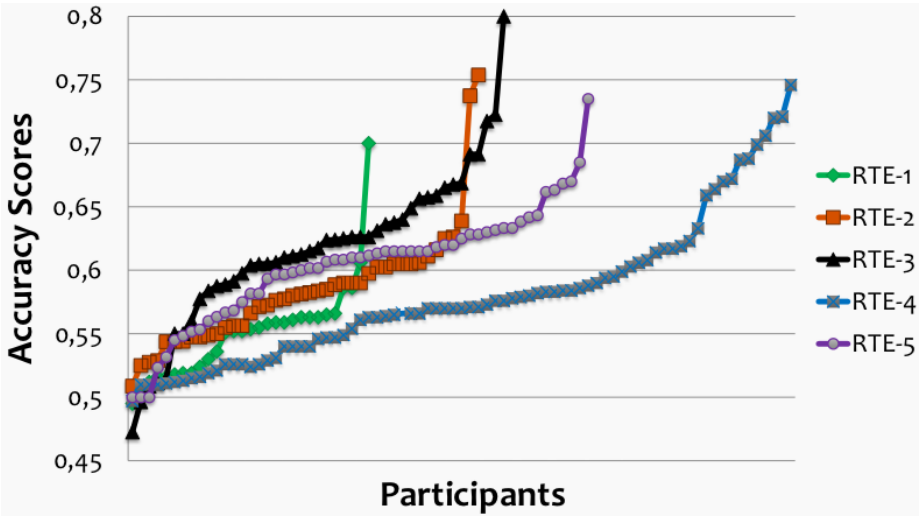

Figure 1: Comparison of accuracies of participants in RTE 1-5

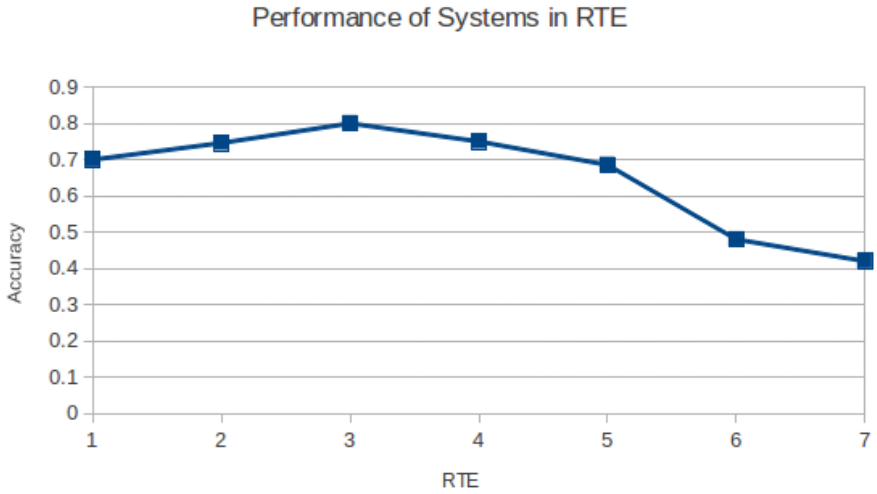

Figure 2: Trend of RTE best scores

re-representation of *hypothesis* is subsumed by that of the *text*. Most of the systems are based on Machine Learning approaches. The entailment decision problem can be considered as a classification problem. Such systems use features such as lexical, syntactic and semantic features.

RTE challenges provided a great platform for Textual Entailment systems and because of it, a wide variety of approaches emerged every year. Some approaches

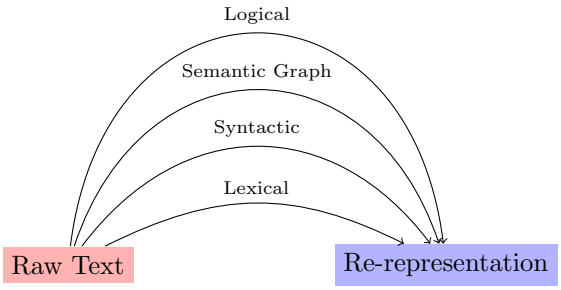

Figure 3: Various Representations

were Machine Learning based approaches such as Supervised Machine Learning approach (Agichtein et al., 2008) and Tri-categorization approach to Textual Entailment (Ren et al., 2009) which use system of classification based on lexical, syntactic and semantic features. These systems use WordNet for semantic features. Probabilistic approaches such as Textual Entailment based on a calculus on dependency parse trees (Harmeling, 2009) and Modeling framework for lexical entailment, with suitable EM-based parameter estimation (Shnarch et al., 2011) were also proposed. Approaches based on tree edit distance algorithms (Kouylekov and Magnini, 2006), (Tatu et al., 2006) and (Bar-Haim et al., 2007) have been used. Heilman *et al.* (2010) propose tree edit models for representing sequences of transformation and employs tree kernel heuristic in a greedy search routine (Heilman and Smith, 2010). Sammons *et al.* (2009) use shallow semantic representation for alignment based approach. An interesting approach for cross-lingual textual entailment is proposed by Mehdad et al. (2010) which uses bilingual parallel corpora. They obtained good results on RTE datasets by using monolingual parallel corpora for English language.

Machine Learning and Probability based approaches are not the only approaches used. Determining the deep semantic inferences from the text was also proposed. Approaches based on logical inferences (Bos and Markert, 2005) and the application of natural logic (MacCartney and Manning, 2007) yielded good accuracy. Recently, work on the use of deep semantic inferences for Recognizing Textual Entailment is going on in IIT Bombay. It uses Universal Networking Language(UNL) graph representation.

4 Lexical Approaches

Lexical approaches work directly on the input surface strings. These approaches generally incorporate some pre-processing, such as part-of-speech (POS) tagging or named-entity recognition (NER). These approaches do not retrieve syntactic or semantic information from the text. Entailment decisions are taken only from the lexical evidences. Common approaches include word overlap, subsequence

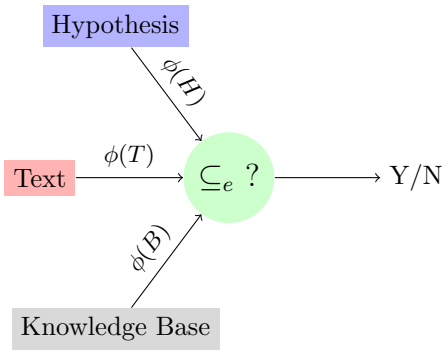

Figure 4: General Strategy

matching, longest substring using sliding window approach *etc.* This chapter explains lexical approaches for recognizing textual entailment. General strategy of lexical approaches is:

1. Pre-process the given texts to separate content words and unwanted words.
2. Re-represent the texts.
3. Compare these re-represented texts for matching
4. Decide entailment based on the matching score.

This strategy is explained in details in the subsequent sections.

4.1 Preprocessing

In lexical approaches, preprocessing step involves *tokenization*, *stemming/ lemmatization* and identifying the *stop words*. Stop words *e.g. a, an, the etc.*, unlike *content words*, do not contribute to recognition of entailment. This is because they occur too frequently to imply any entailment. Certain systems also carry out some deeper pre-processing tasks such as:

- **Phrasal Verb Recognition:** This step identifies all the phrasal verbs in both text and hypothesis. Examples of phrasal verbs are *take off, pick up etc.*
- **Idiom processing:** An idiom is an expression, word, or phrase that has a figurative meaning that is comprehended in regard to a common use of that expression that is separate from the literal meaning or definition of the words of which it is made. There are estimated to be at least 25,000 idiomatic expressions in the English language. Examples of some idioms are:
  - You should **keep an eye out** for that. - *to keep an eye out for something means to watch for it.*

– I knew that Granny was **pulling** my **leg**. - *to pull someone's leg means to tease them by telling them something fictitious.*

Idioms in the form of complete sentences are known as *Proverbs*, if they refer to the universal truth. For example:

– *Well begun is half done.*

– *Nothing ventured, nothing gained.*

– *You can lead a horse to the river, but you can't make him drink.*

Since they mean something different from what they mean, lexical approach would fail. Therefore they are required to be treated separately. In this step, known idioms are identified and are replaced by actual meaning.

- **Named Entity Recognition and Normalization:** Named entities such as name of person, company *etc.* are represented in various forms. This step identify the named entities in text and hypothesis, and normalizes them to some single notation. One approach to normalize is replacing spaces by underscores. For example, *Leonardo DiCaprio* is combined to form *Leonardo_DiCaprio* and *United States of America* is normalized as *United_States_of_America*.

- **Date/Time arguments:** This step is similar to Named Entity Recognition except that it identifies date and time elements.

An example:

T: Eying the huge market potential, currently led by Google, Yahoo **blue**took over search company **blue** Overture Services Inc. last year.

H: Yahoo **acquired Overture**.

In the example **Overture Services Inc.** and **Overture** are normalized by named entity recognition and the phrasal verb **took over** is mapped to **acquired**.

4.2 Representation

After the preprocessing, the text T and the hypothesis H are re-represented, in case of lexical approaches as one of the following:

- Bag-of-words: Both T and H are represented as a set of words.

- n-grams: Sequence of n tokens are grouped together. Bag of words is an extreme case of n-gram, with n=1, known as unigrams.

Example: *Edison invented the light bulb in 1879, providing a long lasting source of light.*

- Bag-of words: {*Edison, invented, the, light, bulb, in, 1879, providing, a, long, lasting, source, of, light*}
- Bigram model (n-gram with n=2): {*Edison invented, invented the, the light, light bulb, bulb in, in 1879, 1879 providing, providing a, a long, long lasting, lasting source, source of, of light.*}

Re-representations of text and hypothesis are then compared with each other to calculate the matching score which decides the entailment. Matching is carried out on the basis of the information obtained with the help of knowledge resources.

4.3 Knowledge Resources

Lexical Approaches typically uses shallow lexical resource such as WordNet. The knowledge resources are used to measure the similarity between re-represented text and hypothesis. Some of the various properties used are:

- **Hyponymy:** Hyponymy denotes the specialization of the concepts. Hyponym relation gives the specific term used to designate a member of a class. X is a hyponym of Y if X is a (kind of) Y. *e.g. Sparrow* is hyponym of *Bird* and *Mumbai* is a hyponym of *City* .
- **Hypernym:** The generic term used to designate a whole class of specific instances. Y is a hypernym of Xif X is a (kind of) Y. It is the reverse relation of Hyponymy. *e.g. Bird* is hypernym of *Sparrow* and *City* is Hypernym of *Mumbai*.
- **Meronymy/Holonymy:** Meronymy is the name of a constituent part of, the substance of, or a member of something. X is a meronym of Y if X is a part of Y. The reverse relation is Holonymy. For example, *Wheel* is a meronym of *Car* and *Orchestra* is a holonym of *Musician*.
- **Troponym:** Relation which denotes *manner-of*. A verb expressing a specific manner of another verb. X is a troponym of Y if to X is to Y in some manner. *Limping* is troponym of *Walk*.
- **Entailment:** A verb X entails Y if X cannot be done unless Y is, or has been, done. *e.g. Snoring* entails *Sleeping*.

4.4 Control Strategy and Decision Making

The lexical approaches employ a *single pass* control strategy. That means unlike iterative methods, they reach the decision in a single iteration. Decision making is done based on a certain threshold (decided experimentally) over the similarity scores generated by the algorithms. The similarity scores are calculated based on WordNet distances using properties mentioned in section 4.3.

```
INPUT: Text T and Hypothesis H.
OUTPUT: The matching score.
for all word in T and H do
  if word in stopWordList then
    remove word;
  end if
  if no words left in T or H then
    return 0;
  end if
end for
numberMatched = 0;
for all word W_T in T do
  Lemma_T = Lemmatize(W_T);
  for all word W_H in H do
    Lemma_H = Lemmatize(W_H);
    if LexicalCompare(Lemma_H, Lemma_T) then
      numberMatched++;
    end if
  end for
end for
```

Figure 5: LLM Algorithm

```
if Lemma_H == Lemma_T then
  return TRUE;
end if
if HypernymDistance(W_H , W_T) ≤ d_Hyp then
  return TRUE;
end if
if MeronymDistance(W_H , W_T) ≤ d_Mer then
  return TRUE;
end if
if MemberOfDistance(W_H , W_T) ≤ d_Mem then
  return TRUE;
end if
if SynonymOf(W_H , W_T) then
  return TRUE;
end if
```

Figure 6: Lexical Compare Procedure

5 Machine Learning Approaches

Entailment problem can be thought of as a **YES/NO** classification problem (figure 7). Given the text T and the hypothesis H, if we are able to use various similarity features between the texts, forming a feature vector, we can use this feature vector to classify the problem of entailment using a standard classifier

(say SVM classifier). Two classes can denote whether the entailment between a pair of text and hypothesis is true (*class YES*) or false (*class No*).

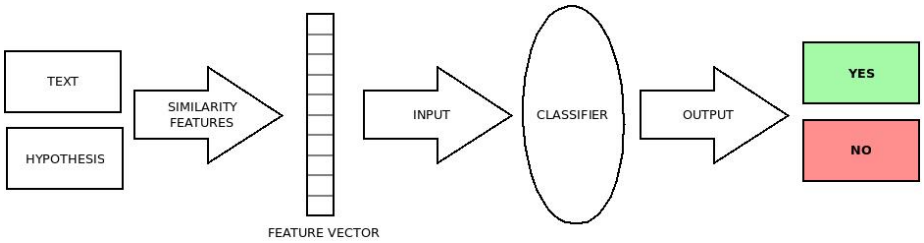

Figure 7: Textual Entailment as a Classification Task

5.1 Feature Space

Machine Learning approaches focus on the prediction, based on the properties learnt from the training data. It is of utmost importance to determine the feature space for the given problem and the corresponding training data. In case of the entailment problem, possible feature space can be (Inkpen et al., 2006):

- **Distance Features** Features of some distance between T and H. Smaller distances mean that the texts are lexically, syntactically or semantically closer.
- **Entailment Triggers** Features that triggers entailment (or non-entailment)
- **Pair Feature** Content of T-H pair

5.2 Distance Features

The distance features models the distance between the text and the hypothesis in some way (Zanzotto et al., 2009). Machine Learning approaches can use lexical features, too. For example:

- Number of words in common
- Length of longest common subsequence
- Longest common syntactic sub-tree

For example:

T: *HDFC Bank, India's No.3 lender, met forecasts with a 30 percent rise in quarterly profit, led by stronger loan growth, better fee income and stable net interest margins.*

H: *HDFC Bank, expected a 30 percent rise in quarterly profit, led by stronger loan growth, better fee income and firm net interest margins.*

The above example, possible $\langle feature, value \rangle$ pair could be $\langle WordsInCommon, 21 \rangle$ or $\langle LongestSubsequence, 16 \rangle$.

5.3 Entailment Triggers

Entailment triggers can be used as possible features for classifiers. Some of them are:

Polarity Features: Presence or absence of negative polarity. Since presence of the same polarity in both the *text* and *hypothesis* may lead to entailment.

Example: Rupee was *down* by 5 paise against dollar in early trade. $\models$ Rupee *did not rise* against dollar.

Antonymy Features: Presence or absence of antonymous words in T and H.

Example: Bank was *close* on that day. $\not\models$ Bank was *open* on that day.

Adjunct Features: Dropping/adding of syntactic adjunct.

Example Sunny goes to school. $\models$ Sunny goes to school *regularly.*

5.4 Pair Features

In this feature space, we try to model the content of T and H rather than modeling the distance between them. While using this feature space, choosing the right features is crucial. Following example illustrates why it could be bad if we select wrong features. Lets take an example to explain the effective use of pair feature space. Consider

T: *At the end of the year, all solid companies pay dividends.*

H_1**:** *At the end of the year, all solid insurance companies pay dividends.*

H_2**:** *At the end of the year, all solid companies pay cash dividends.*

If we would have taken distance feature, it would plot both $\langle T, H_1 \rangle$ and $\langle T, H_2 \rangle$ to be same point in feature space. What we need is something that can model the *content* and the *structure* of the T and H. An approach is presented (Zanzotto et al., 2009) where each T-H pair is projected to a vector that, roughly speaking, contains as features all the fragments of parse trees of T and H.

5.4.1 The Kernel Trick

To solve the above problem, we use a syntactic pair feature space (Bar-Haim et al., 2007). To do this, instead of taking the features separately, we use kernels to represent the distance between two example pairs.

Cross Pair Similarity:

$$K(< T', H' >, < T'', H'' >) = K(< T', T'' >) + K(< H', H'' >)$$

We desire the definition of $K(P_1, P_2)$ to be such that it can exploit the *rewrite rules* of the examples. For this, placeholders were introduced in the syntactic tree. The cross pair similarity is based on the distance between syntactic trees with *co-indexed leaves*:

$K(< T', H' >, < T'', H'' >) =$

$$\max_{c \in C}(K_T(t(H', c), t(H'', i)) + K_T(t(T', c), t(T'', i))) \quad (2)$$

where,

C is the set of all correspondences between anchors of (T',H') and (T'',H'')

$t(S, c)$ renames the anchors in the parse tree S with configuration c.

i is the identity mapping

$K_T(t_1, t_2)$ is a similarity measure between t_1 and t_2

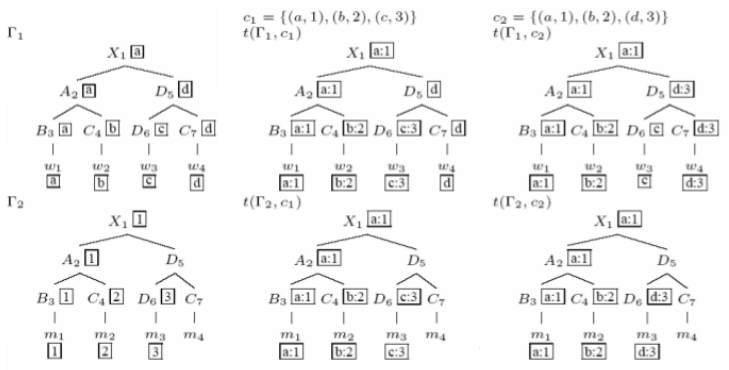

Figure 8: Example of cross pair similarity between two parse trees.

Following example illustrates the above concept. Figure 8 shows cross pair similarity between two parse trees Γ_1 and Γ_2 with the placeholders according to two configurations c_1 and c_2. The configuration is selected using the definition of $K(P_1, P_2)$.

How the rewrite rules are exploited is illustrated by following example. Consider the the pair:

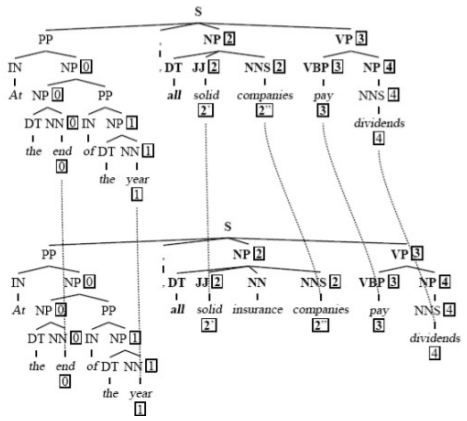

Figure 9: Intra-pair similarity after propagating the placeholders.

T: *Chapman killed Lennon.*

H: *Lennon died.*

Using the syntactic pair features and the kernel representation described above we can learn useful rules (unlike those learned using *bag of words*) such as in figure 10.

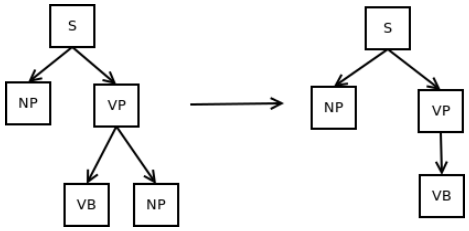

Figure 10: Exploiting rewrite rules

5.5 Classifier Algorithms

As discussed above, selection suitable features is the main problem in machine learning based approaches. Once done, any off-the-shelf classifier can be used for the classification tasks.

Using syntactic pair feature kernel and SVM, the accuracy of the system on RTE 3 data set was 62.97%. Using the approach together with lexical distance features, however, raises the accuracy up to 68.26 (Zanzotto et al., 2009).

6 Approaches based on Graph Matching

Bag-of-words or n-gram model representation can take us only so far. For deeper understanding of the sentences we eventually would require to show how the words in the sentence affects each other, i.e. how are the words dependent on each other. These dependencies could be syntactic (e.g. which phrase does the word belong) or semantic (e.g. what role does the word play). This chapter explores how such representation could help in the task of textual entailment.

6.1 TE as Graph Matching Problem

Textual entailment can be seen as graph matching problem. In this approach, we convert the hypothesis H and text T into graphs. There are various ways, syntactic or semantic, to convert a natural language sentence to graph. Once we get the text and hypothesis graph, its about finding subsumption of sub-graphs. We apply some graph matching techniques to determine the matching score between the two graphs. If the score attains a certain threshold, entailment is labeled as valid.

6.2 Comparison with Classical Graph Matching

Although the problem described above seems like a straightforward sub-graph matching problem, we can not use existing concepts of determining match between sub-graphs. Here are the reasons:

- The scoring in Textual Entailment is **not** symmetric. Score between H and T is not same as that between T and H. Classical graph matching is however, symmetric.
- Linguistically motivated graph transformation (nominalization, passivization) are to be considered in case of textual entailment. So, unlike classical graph matching, we can not measure similarity at the node level.

6.3 Conversion from Natural Language Text to Graph

Here we illustrate the procedure to convert a natural language text to a dependency graph. Starting with raw English text, a parser is used to obtain a parse tree. Then, a dependency tree representation of the sentence was derived using a slightly modified version of Collins' head propagation rules (Collins, 1999), which make main verbs and not auxiliaries the head of sentences. Edges in the dependency graph are labeled by a set of hand-created rules. These labels represent "surface" syntax relationships such as *subj* for subject and *amod* for adjective modifier. The dependency rules are created as follows:

- Take each rule $X \rightarrow Y_1...Y_n$ such that:
    1. $Y_1, ..., Y_n$ are non terminals.

2. $n \geq 2$.
3. $h = head(X \rightarrow Y_1...Y_n)$.

- Each rule contributes $n-1$ dependencies, $headword(Y_i) \rightarrow headword(Y_h)$ for $1 \leq i \leq n$, $i \neq h$.
- if X is a root non terminal and x is its headword, then $x \rightarrow START$ is a dependency.

For example, consider the sentence:

Workers dumped sacks into a bin.

The dependencies are:

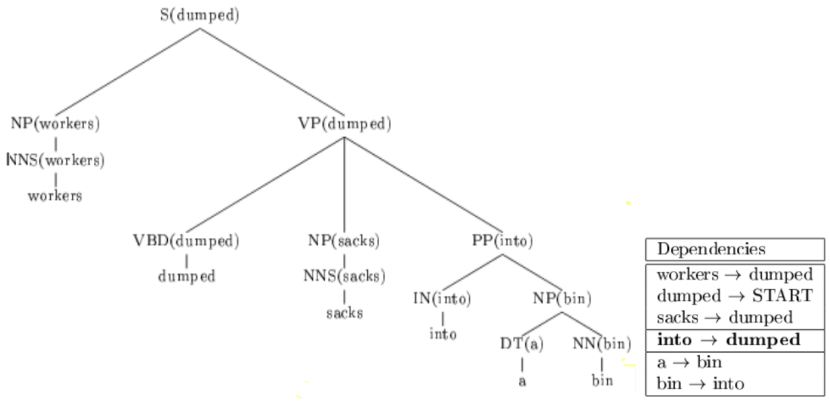

Figure 11: Parse tree and extracted dependencies

6.4 Enhancements to Dependency Graphs

Using the above dependency graph as the base, various enhancements can be applied to the graphs (Haghighi et al., 2005).

1. **Collapse Collocations and Named-Entities:** We *collapse* dependency nodes which represent named entities (e.g. nodes [Swapnil] and [Ghuge] could be collapsed into [Swapnil Ghuge]) and also collocations listed in WordNet, including phrasal verbs (e.g., *blow off* in *He blew off his work*).

2. **Dependency Folding:** It was found that it is useful to fold certain dependencies (e.g. modifying prepositions such as "in", "'under" etc.) so that modifiers became labels connecting the modifier's governor and dependent directly.

3. **Semantic Role Labeling:** Graph representation was augmented with Probank-style semantic roles. Each predicate adds an arc labeled with the appropriate semantic role to the head of the argument phrase. This helps to create links between words which share a deep semantic relation not evident in the surface syntax. Additionally, modifying phrases are labeled with their semantic types (e.g., *Pakistan got independence in* $[1947]_{Temporal}$.), which should be useful in Question Answering tasks.

4. **Co-reference links:** Using a co-resolution tagger, *coref* links are added throughout the graph. These links, connecting referring and referent entities, are the only link between two sentences.

6.5 The Entailment Model

For hypothesis graph H, and text graph T, a matching M is a mapping from the vertices of H to those of T. For vertex v in H, $M(v)$ denotes its *match* in T.

As is common in statistical machine translation, nodes in H are allowed to map to fictitious $NULL$ vertices in T if necessary.

Suppose the cost of matching M is $Cost(M)$. If $\mathbf{M}$ is the set of such matchings, the cost of matching H to T is defined to be:

$$MatchCost(H,T) = \min_{M \in \mathbf{M}} Cost(M) \tag{3}$$

Let $VertexSub(v, M(v))$ be a model which gives us a cost in [0, 1], for substituting vertex v in H for $M(v)$ in T. Then,

$$VertexCost(M) = \frac{1}{Z} \sum_{v \in H_v} w(v) * VertexSub(v, M(v)) \tag{4}$$

where $w(v)$ is relative importance of vertex v, and Z is the normalizer, $\sum_{allv} w(v)$. Now, consider an edge $e = (v, v') \in H_E$, and let $\phi_M(e)$ be the path from $M(v)$ to $M(v')$ in T. Let $PathSub(e, \phi_M(e))$ be a model for assessing the *cost* of substituting a direct relation $e \in H_E$ for its counterpart, $\phi_M(e)$, under the matching. This leads to formulation of $RelationCost(M)$ in a similar fashion:

$$RelationCost(M) = \frac{1}{Z} \sum_{e \in H_E} w(e) * PathSub(e, \phi_M(e)) \tag{5}$$

The total cost function could then be expressed as a convex combination of the two cost functions as:

$$Cost(M) = \alpha * VertexCost(M) + (1 - \alpha) * RelationCost(M) \tag{6}$$

6.6 Vertex Substitution Cost Model

The vertex substitution cost model is based on following factors.

- **Exact Match:** v and $M(v)$ are identical words/phrases.
- **Stem Match:** v and $M(v)$'s stems match or one is a derivational form of the other; e.g., matching *coaches* to *coach*
- **Synonym Match:** v and $M(v)$ are synonyms according to WordNet.
- **Hypernym Match:** v is a *kind of* $M(v)$, as determined by WordNet. Note that this feature is asymmetric.
- **WordNet Similarity:** v and $M(v)$ are similar according to WordNet::Similarity (e.g. low conceptual distance).
- **LSA Match:** v and $M(v)$ are distributionally similar according to a freely available Latent Semantic Indexing package, or for verbs similar according to VerbOcean.
- **POS Match:** v and $M(v)$ have the same part of speech.
- **No Match:** $M(v)$ is $NULL$.

6.7 Path Substitution Cost Model

Similar to the Vertex Substitution Cost Model, we define Path Substitution Cost Model based on following factors.

- **Exact Match:** $M(v) \rightarrow M(v')$ is an en edge in T with the same label.
- **PartialMatch:** $M(v) \rightarrow M(v')$ is an en edge in T, not necessarily with the same label.
- **AncestorMatch:** $M(v)$ is an ancestor of $M(v')$. An exponentially increasing cost is used for longer distance relationships.
- **KinkedMatch:** $M(v)$ and $M(v')$ share a common parent or ancestor in T. An exponentially increasing cost is used based on the maximum of the node's distances to their least common ancestor in T.

6.8 Additional Checks

Certain additional checks can be applied to the system to improve its performance (Haghighi et al., 2005). They are listed below.

- **Negation Check:** Check if there is a negation in a sentence. Example,
  - **T:***Clinton's book is not a bestseller.*
  - **H:***Clinton's book is a bestseller.*
- **Factive Check:** Non-factive verbs (claim, think, charged, etc.) in contrast to factive verbs (know, regret, etc.) have sentential complements which do not represent true propositions.

- **T:** *Clonaid claims to have cloned 13 babies worldwide.*
- **H:** *Clonaid has cloned 13 babies.*

- **Superlative Check:** invert the typical monotonicity of entailment. Example,
  - **T:** *The Osaka World Trade Center is the tallest building in Western Japan.*
  - **H:** *The Osaka World Trade Center is the tallest building in Japan.*
- **Antonym Check:** It is observed that the WordNet::Similarity measures gave high similarity to antonyms. Explicit check of whether a matching involved antonyms is done and unless one of the vertices had a negation modifier, its rejected.

7 Semantics-based Approaches

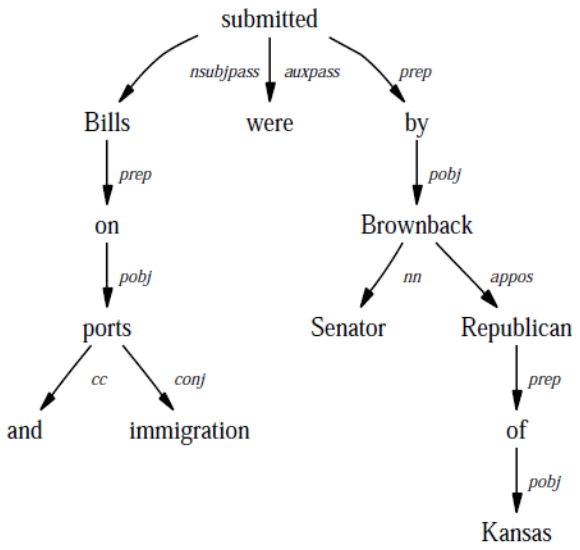

Figure 12: The Stanford dependency graph for *Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas*

Semantics-based approaches differ from those discussed earlier, in a sense, that these approaches actually consider the meaning of the texts. These approaches map language expressions to semantic representations. Some of the approaches map language expressions to logical meaning representations (Tatu and Moldovan, 2005). Semantics based approaches rely heavily on knowledge resources such as

WordNet, VerbOcean, VerbNet, DIRT, *etc.* In these approaches, semantics are generally represented in a graphical structure.

Graph based matching algorithms were proposed by Haghighi et al. (2005), Herrera et al. (2006). Natural logic based approaches like (MacCartney and Manning, 2007) represent meaning of the text in the form of logic expressions and then determine the truth value of the *hypothesis*. Other semantics based approaches are mentioned in the summary.

8 Summary

Challenging evaluation forum like RTE included examples which not only required lexical match but also needed semantic relations to determine the entailment. From various examples, it was clear that simple surface string matching or syntactic overlapping is not sufficient to recognize entailment. As a result, deep semantic approaches emerged. But extracting fine grained information from the text is a difficult task. Hence deep semantic based approaches are also not very robust. With the following tables mentioning various approaches, we conclude our survey about textual entailment.

Approaches	**Description**	**Results**
Bilingual Corpora (Mehdad et al., 2011)	Use of bilingual parallel corpora as a lexical resource for cross-lingual text entailment.	RTE-3: Avg Acc=62.37% RTE-5: Avg Acc=61.41%
Probabilistic (Shnarch et al., 2011)	Probabilistic modeling framework for lexical entailment.	RTE-5: F=44.7% RTE-6: F=45.5%
Machine Learning (Mehdad et al., 2009)	Use of semantic knowledge based on Wikipedia, used to enrich the similarity measure between pairs of text and hypothesis.	RTE-5: Acc=66.2%, Prec=66%
Shallow Semantic approach (Sammons et al., 2009)	Shallow semantic representation. Alignment based approach is proposed.	RTE-5 Dev: Acc=66.7%, Test: Acc=67%

Table 2: Recent work in the field of Text Entailment - Part 3

Approaches	Description	Results
Machine Learning (Ren et al., 2009)	System of classification which considers lexical, syntactic and semantic features. For semantic features, WordNet is used.	RTE-5, 63.3%
Machine Learning (Agichtein et al., 2008)	Supervised Machine Learning approach. Use WordNet for semantic similarity, NLTK to find path distances.	RTE-4, Acc=58.8%, Prec=60.0%
Probabilistic (Harmeling, 2009)	Probabilistic Model of Entailment.	RTE-2, training: Acc=62.5%, Prec=65.51% RTE-3, training: Acc=62.12%, Prec=63.12%
Discourse Extraction (Hickl, 2008)	New framework depends on extracting a set of publicly-held beliefs - known as discourse commitments.	RTE-2 and 3: Accuracy 84.93%
Syntactic similarity (Androutsopoulos and Malakasiotis, 2010)	Capture similarities at various abstractions of the input. Use WordNet and features at syntactic level.	RTE-1 dataset: Acc=52.88%, Prec=52.77% RTE-2 dataset: Acc=57.88%, Prec=56.5% RTE-3 dataset: Acc=64.63%, Prec=63.26%
Tree Edit Distance	Tree edit models for representing sequences of tree transformations. To efficiently extract sequences of edits, they employ a tree kernel heuristic in a greedy search routine.	RTE-3: Acc=62.8%, Prec=61.9%
Tree Kernel based approach (Mehdad et al., 2010)	Based on off-the-shelf parsers and semantic resources for recognizing textual entailment. Syntax is exploited by means of tree kernels while semantics is derived from WordNet, Wikipedia etc.	RTE-2: Avg Acc=59.8%, RTE-3: Avg Acc=64.5%, RTE-5: Avg Acc=61.5%

Table 3: Recent work in the field of Text Entailment - Part 2

References

Eugene Agichtein, Walt Askew, and Yandong Liu. Combining lexical, syntactic, and semantic evidence for textual entailment classification. In *Text Analysis*

Conference, 2008.

Ion Androutsopoulos and Prodromos Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187, 2010. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.167.4426&rep=rep1&type=pdf.

R. Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, pages 1–9. Citeseer, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.9327&rep=rep1&type=pdf.

R. Bar-Haim, Ido Dagan, I. Greental, and E. Shnarch. Semantic inference at the lexical-syntactic level. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22, page 871. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Semantic+Inference+at+the+Lexical-Syntactic+Level#0.

Luisa Bentivogli, Danilo Giampiccolo, Hoa Trang Dang, Ido Dagan, and Bernardo Magnini. The fifth recognizing textual entailment challenge overview. In *Text Analysis Conference*, 2009.

Arindam Bhattacharya. Recognizing textual entailment using deep semantic graphs and shallow syntactic graphs. Technical report, Indian Institute of Technology, Bombay, 2012.

Johan Bos and Katja Markert. Recognising textual entailment with logical inference. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pages 628–635, 2005. doi: 10.3115/1220575.1220654. URL http://portal.acm.org/citation.cfm?doid=1220575.1220654.

Micheal Collins. Head-driven statistical models for natural language parsing. *Ph.D. thesis, University of Pennsylvania.*, 1999.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, 2005. URL http://www.springerlink.com/index/D28U2080M6532524.pdf.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The Fourth PASCAL recognizing textual entailment challenge. *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing - RTE '07*, 2007a. doi: 10.3115/1654536.1654538. URL http://portal.acm.org/citation.cfm?doid=1654536.1654538.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, number June, pages 1–9, Morristown, NJ, USA, 2007b. Association for Computational Linguistics. doi: 10.3115/1654536.1654538. URL http://portal.acm.org/citation.cfm?doid=1654536.1654538http://portal.acm.org/citation.cfm?id=1654538.

Oren Glickman, Ido Dagan, and Moshe Koppel. A probabilistic lexical approach to textual entailment. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 19, page 1682. Citeseer, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.2503&rep=rep1&type=pdf.

Aria D. Haghighi, Andrew Y. Ng, and Christopher D. Manning. Robust textual inference via graph matching. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pages 387–394, 2005. doi: 10.3115/1220575.1220624. URL http://portal.acm.org/citation.cfm?doid=1220575.1220624.

Stephen Harmeling. Inferring textual entailment with a probabilistically sound calculus. *Natural Language Engineering*, (15):459–477, 2009.

Michael Heilman and Noah A. Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. *Human Language Technologies*, pages 1011–1019, 2010.

J. Herrera, A. Penas, and Felisa Verdejo. Textual entailment recognition based on dependency analysis and wordnet. *Machine Learning Challenges*, pages 231–239, 2006. URL http://www.springerlink.com/index/m44732m5mp780250.pdf.

Andrew Hickl. Using discourse commitments to recognize textual entailment. In *Proceedings of the 22nd International Conference on Computational Linguistics Volume 1*, volume 1 of *COLING '08*, pages 337–344. Association for Computational Linguistics, Association for Computational Linguistics, 2008. ISBN 9781905593446. doi: 10.3115/1599081.1599124. URL http://portal.acm.org/citation.cfm?id=1599081.1599124.

Andrew Hickl and Jeremy Bensley. A discourse commitment-based framework for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing - RTE '07*, pages 171–179, Morristown, NJ, USA, 2007. Association for Computational Linguistics. doi: 10.3115/1654536.1654571. URL http://portal.acm.org/citation.cfm?doid=1654536.1654571.

Andrew Hickl, John Williams, Jeremy Bensley, and Kirk Roberts. Recognizing textual entailment with LCC's GROUNDHOG system. *Proceedings of the*

Recognising Textual Entailment Challange, 2006. URL http://u.cs.biu.ac.il/~nlp/RTE2/Proceedings/14.pdf.

Diana Inkpen, Darren Kipp, and Vivi Nastase. Machine learning experiments for textual entailment. *Proceedings of the second RTE Challenge, Venice-Italy*, 2006. URL http://www.site.uottawa.ca/~{}dkipp/pub/entailment2006.pdf.

Milen Kouylekov and Bernardo Magnini. Tree edit distance for recognizing textual entailment: Estimating the cost of insertion ber. In *PASCAL RTE-2 Challenge*, 2006.

Bill MacCartney and Christopher D. Manning. Natural logic for textual inference. *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing - RTE '07*, pages 193–201, 2007. doi: 10.3115/1654536.1654575. URL http://portal.acm.org/citation.cfm?doid=1654536.1654575.

Yashar Mehdad, Alessandro Moschitti, and Fabio Massimo Zanzotto. Semker: Syntactic/semantic kernels for recognizing textual entailment. In *Text Analysis Conference*, 2009.

Yashar Mehdad, Alessandro Moschitti, and Fabio Massimo Zanzotto. Syntactic/semantic structures for textual entailment recognition. In *Ascociation of Computational Linguistics*, 2010.

Yashar Mehdad, Matteo Negri, and Marcello Federico. Using bilingual parallel corpora for cross-lingual textual entailment. In *49th Annual Meeting of the Association for Computational Linguistics*, pages 1336–1346, 2011.

Han Ren, Donghong Ji, and Jing Wan. A tri-categorization approach to textual entailment recognition. In *Text Analysis Conference*, 2009.

Mark Sammons, V. G. Vinod Vydyswaran, et al. Relation alignment for textual entailment recognition. In *Text Analysis Conference*, 2009.

Eyal Shnarch, Jacob Goldberger, and Ido Dagan. A probabilistic modeling framework for lexical entailment. In *49th Annual Meeting of the Association for Computational Linguistics*, pages 558–563, 2011.

Marta Tatu and Dan Moldovan. A semantic approach to recognizing textual entailment. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, (October 2005):371–378, 2005. doi: 10.3115/1220575.1220622. URL http://portal.acm.org/citation.cfm?doid=1220575.1220622.

Marta Tatu, Brandon Iles, John Slavick, Adrian Novischi, and Dan Moldovan. Cogex at the second recognizing textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 104–109, 2006. URL http://u.cs.biu.ac.il/~{}nlp/RTE2/Proceedings/17.pdf.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. A machine learning approach to textual entailment recognition. *Natural Language Engineering*, 15(04):551–583, September 2009. ISSN 1351-3249. doi: 10.1017/S1351324909990143. URL http://www.journals.cambridge.org/abstract_S1351324909990143.