Unsupervised Resource Creation for Textual Inference Applications

Jeremy Bensley and Andrew Hickl

Language Computer Corporation
1701 North Collins Boulevard Suite 2000
Richardson, Texas USA
{jeremy,andy}@languagecomputer.com

Abstract

This paper explores how a battery of unsupervised techniques can be used in order to create large, high-quality corpora for textual inference applications, such as systems for recognizing textual entailment (TE) and textual contradiction (TC). We show that it is possible to automatically generate sets of positive and negative instances of textual entailment and contradiction from textual corpora with greater than 90% precision. We describe how we generated more than 1 million TE pairs – and a corresponding set of 500,000 TC pairs – from the documents found in the 2 GB AQUAINT-2 newswire corpus.

1. Introduction

The emergence of robust, machine learning-based approaches to the recognition of *textual entailment* and *textual contradiction* has underscored the need for large sources of training data which can be used to construct accurate models for recognizing textual inference.

First described in (Glickman and Dagan, 2005), the task of recognizing textual entailment (RTE) requires systems to determine whether a short statement (conventionally known as a *hypothesis* (or *h*)) can be conventionally inferred from a longer passage (known as a *text* (or *t*)). (Figure 1 presents both a positive and negative instance of TE.[1])

TE	Example
YES	**Text:** Indian firm Tata Steel has won the battle to take over Anglo-Dutch steelmaker Corus.
	Hypothesis: Tata Steel bought Corus.
NO	**Text:** Dynamite Nobel is formed by the fusion of Nobel's Italian and Swiss companies.
	Hypothesis: Alfred Nobel is the inventor of dynamite.

Table 1: Examples of Textual Entailment

Following the success of the PASCAL RTE evaluations (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007), (Harabagiu et al., 2006) introduced a complementary form of inference, known as *textual contradiction* (TC). In (Harabagiu et al., 2006)'s framework, a t is considered to textually contradict a h if there exists any proposition inferable from t which could lead to the refutation of h. (Figure 2 presents positive and negative instances of textual contradiction.)

TC	Example
YES	**Text:** The explosion wounded the arm of Beatriz Iero, damaged the doors and walls of the offices, and broke the windows of neighboring buildings.
	Hypothesis: Beatriz Iero emerged unscathed from an explosion
NO	**Text:** In California, one hundred twenty Central Americans, due to be deported, began a hunger strike when their deportation was delayed.
	Hypothesis: The deportation of 120 Central Americans was postponed.

Table 2: Examples of Textual Contradiction

The recognition of forms of textual inference such as TE and TC has traditionally been considered the domain of formal, logic-based methods (such as automatic theorem proving- (Tatu et al., 2006)) or model-based approaches (such as model building or model checking (Blackburn and Bos, 2005)). However, a considerable amount of recent work – including many of the top-performing systems at the past PASCAL RTE Challenges (Hickl and Bensley, 2007; Hickl et al., 2006; Haghighi et al., 2005) – has demonstrated the effectiveness of using "shallow" statistical classifiers in order to recognize TE (or TC) relations. While individual systems have exploited a wide range of different types of features in order to perform this classification (including syntactic heuristics (Vanderwende et al., 2006), graph matching techniques (Raina et al., 2005), or output from model checking (Bos and Markert, 2006) or paraphrasing (Hickl et al., 2006) applications), access to sources of training data has continued to be a limiting factor.

This paper follows initial work done by (Burger and Ferro, 2005; Brockett and Dolan, 2005; Dolan and Quirk, 2004) in exploring how a battery of unsupervised techniques can be used in order to create large, high-quality corpora for textual inference applications. We show that it is possible to automatically generate sets of positive and negative instances of textual entailment and contradiction from textual corpora with greater than 90% precision. In our work, we describe how we generated more than 1 million TE pairs – and a corresponding set of and 500,000 TC pairs – from the documents found in the 2 GB AQUAINT-2 newswire corpus.

This paper also investigates the impact that sources of generated training data can have on the performance of state-of-the-art systems for recognizing textual entailment (RTE) (Hickl and Bensley, 2007) and recognizing textual contradiction (RTC) (Harabagiu et al., 2006). Our results confirm the hypothesis (first suggested in (Hickl et al., 2006)) that the performance of classification-based systems for RTE increases with the amount of available training data. In our experiments, increases in accuracy are observed when training on as many as 500,000 inference pairs; performance remains constant (or suffers slight degradation) with larger training corpora. In our experiments, we observed no significant difference in performance when equivalent number of hand-crafted or automatically-generated examples were used to train clas-

[1] Both examples are taken from the PASCAL RTE-3 Test Set. For more information on the PASCAL RTE Challenges, see http://www.pascal-network.org/Challenges/.

sifiers for recognizing textual entailment or textual contradiction.

The rest of this paper is organized in the following way. Section 2 provides an overview of the general learning-based framework for recognizing instances of textual entailment and textual contradiction previously described in (Hickl and Bensley, 2007; Harabagiu et al., 2006; Hickl et al., 2006). Sections 3 and 4 presents the techniques we used to create training corpora for our RTE and RTC systems: Section 3 discusses how we extended extraction-based techniques (similar to those first proposed in (Burger and Ferro, 2005)) for this task, while Section 4 examines how a generative approach can be used to create training pairs which can be used with either a textual entailment or textual contradiction system. Section 5 explores the impact of these sources of training data on the performance of state-of-the-art RTE and RTC systems, while Section 6 presents our conclusions.

2. Learning Textual Inference Relationships

Recognizing whether the information expressed in a h can be inferred from – or contradicted by the information expressed in a t can be cast either as (1) a classification problem or (2) a formal textual inference problem, performed either by theorem proving or model checking. While these approaches apply radically different solutions to the same problem, both methods involve the translation of natural language into some sort of suitable meaning representation, such as real-valued features (in the case of classification), or axioms or models (in the case of formal methods).

We argue that performing this translation necessarily requires systems to acquire forms of (linguistic and/or real-world) knowledge which may not be derivable from the surface form of a t or h. In order to acquire forms of linguistic knowledge for recognizing textual entailment and textual contradiction, we have developed a novel framework which depends on the extraction of *discourse commitments* from a text-hypothesis pair. Following (Gunlogson, 2001; Stalnaker, 1979), we assume discourse commitments represent the set of propositions which can necessarily be inferred to be true given a conventional reading of a text. Formally, we assume that given a commitment set $\{c_t\}$ consisting of the set of discourse commitments inferable from a text t and a hypothesis h, we define the task of recognizing forms of textual inference as a search for the commitment $c \in \{c_t\}$ which maximizes the likelihood that c participates in a particular textual inference relationship with h. (Examples of commitments that can be extracted from a positive instance of TE are presented in Figure 2.)

In our architecture (illustrated in Figure 1), discourse commitments are first extracted from both the t and the h using the approach described in (Hickl and Bensley, 2007).[2] Commitments are extracted from each t and h using an implementation of the probabilistic finite-state transducer (FST)-based extraction framework described in (Eisner, 2002; Eisner, 2003). Given a syntactically and semantically-parsed input string, our system returns a series of output representations which can be mapped (given a set of generation heuristics) to natural language sentences which represent each of the individual commitments which can be extracted from that string. Commitments were extracted using a series of weighted regular expressions; weights were learned for each regular expression using our implementation of (Eisner, 2002). After each candidate commitment was processed by the FST, the natural language form of each returned commitment was then resubmitted to the FST for additional round(s) of extraction until no additional commitments could be extracted from the input string.

Once commitment sets have been extracted for the t and the h, we then use a *commitment selection* module in order to perform a term-based alignment of each commitment extracted from the t against each commitment extracted from the h. We assume that the alignment of two discourse commitments can be cast as a maximum weighted matching problem in which each pair of words (t_i,h_j) in an commitment pair (c_t,c_h) is assigned a score $s_{ij}(t, h)$ corresponding to the likelihood that t_i is aligned to h_j. As with (Taskar et al., 2005b), we use the large-margin structured prediction model introduced in (Taskar et al., 2005a) in order to compute a set of parameters w (computed with respect to a set of features f) which maximize the number of correct alignment predictions ($\bar{y}_i$) made given a set of training examples (x_i).

The top-ranked pair of commitments (c_{t_i}, c_{h_i}) is then sent to an *inference computation* module which estimates the likelihood that the selected c_{t_i} textually entails (or contradicts) the c_{h_i} (and by extension, the likelihood that t textually entails/contradicts h). Commitment pairs are considered in ranked order until a positive judgment is returned, or until no more commitments above a threshold remain. Following work done by many participants in the PASCAL RTE Challenges, we used a decision tree (C5.0 (Quinlan, 1998)) to estimate the likelihood that a commitment pair represented a valid instance of textual entailment or textual contradiction.

In previous work (Hickl et al., 2006), we showed that performance on the RTE task could be increased by more than 10% when a baseline classification-based system was allowed to train on more than 200,000 examples of textual entailment that were heuristically extracted from documents downloaded from the Internet. Although this was an encouraging result, later work revealed that performance degraded significantly on the PASCAL RTE Test Set when the system was trained on “smaller” datasets consisting of between 800 and 2400 examples, roughly the size of the manually-constructed training corpora made available by the PASCAL RTE organizers.

In our current work, we plan to explore how automatic techniques can be leveraged in order to provide sources of training data for RTE and RTC systems which are as good – if not better than – the manually-created sources of training data provided by the PASCAL RTE organizers or by the authors of (Harabagiu et al., 2006)[3]. It is our expectation

[2] Full details of our framework for recognizing instances of TE can be found in (Hickl and Bensley, 2007). Full details of our system for recognizing TC can be found in (Harabagiu et al., 2006).

[3] The PASCAL RTE organizers have released a collection of

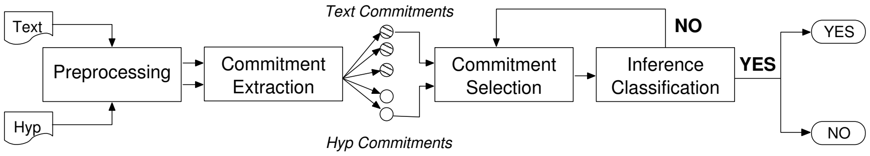


Figure 1: Generic Learning-based Architecture for Recognizing Textual Inference.

that by identifying techniques which are likely to generate valid positive and negative instances of TE/TC, we can assemble sources of training data which will allow for the creation of accurate inference models – even when only a small amount of data is used.

We have grouped methods for generating training data for inference applications into two categories. Section 3 discusses the extractive methods we have developed to find "naturally occurring" entailment or contradiction pairs in text, while Section 4 presents a set of generative methods capable of creating new entailment pairs from complex text passages.

3. Extractive Resource Creation Methods

In this section, we describe how we took advantage of common language constructs and tendencies in order to extracted sets of both positive and negative instances of TE and TC. These methods are derived from analyzing the ways that elaboration, contrast, and paraphrase are generally presented in professional journalism texts.

Our first extractive method follows (Burger and Ferro, 2005) in creating positive textual entailment *t-h* pairs by pairing the first sentence of a newswire document (assumed to be the *t*) with its corresponding headline (assumed to be the *h*). Since the first sentence of a document tends to be an elaboration upon the headline, it follows that most pairs built with this technique contain corresponding information and are positive entailment examples. To prevent the creation of spurious pairings involving uninformative initial sentences, we exclude pairs in which the leading sentence shares no named entity[4] mentions with the headline. A sample analysis performed by human annotators judged that 2296 out of 2500 (91.8%) random pairs generated from this method were positive entailments. (An example of one extracted *t-h* pair is presented in Table 3.)

TE	Example
YES	**Text:** The NCAA on Wednesday named a panel of scientists and sports experts to study the risks associated with metal baseball bats.
	Hypothesis: NCAA Panel To Study Metal Bats

Table 3: Positive Example

Next, we assembled positive instances of textual contradiction by extracting pairs of sentences (or clauses) linked by contrastive discourse connectives such as *although, even though, in contrast, otherwise* and *but*. These language constructs are explicit markers of contradiction or juxtaposition between nuggets of information, and we found that pairs built in this manner have a very high probability of meeting the minimum criteria for textual contradiction. This technique yields significantly fewer samples than other syntactic methods since these discourse connectives appear less frequently in text. Given a random sample of 1000 pairs, human annotators deemed that 94.2% (942) were judged to be valid instances of textual contradiction. (An example of a valid instance of TC is presented in 4.)

TC	Example
YES	**Text:** The sender claimed the letter had a hazardous substance on it, and the office was evacuated [*but*]
	Hypothesis: It appeared there was nothing really wrong with the letter, Police Sgt. Bruce Elrod said.

Table 4: Contradiction Example

In order to provide a large, balanced training set we also needed a mechanism to assemble inference pairs which represented negative instances of TE and TC. In order to create pairs in which the *t* and the *h* expressed similar information, yet could not be considered to be instances of TE or TC, we selected pairs of sentences from an individual document that featured a full mention the same named entity. We assume that sequential entity mentions in a document will convey distinct bits of information with very little redundancy while still making reference to the same topical content, and therefore will almost always represent negative instances of textual entailment or textual contradiction. In our evaluations, human annotators determined that 85.5% of the 2500 random pairs did not meet the minimum criteria to be considered valid instances of TE or TC. (An example of one generated pair is presented in Table 5.) The yield of this technique is significantly higher than the other extraction-based methods since sequential sentences containing mentions of the same entity constitute the majority of news articles.

TE /TC	Example
NO	**Text:** The Steelers were interested in signing Hostetler before the 1997 season–Stewart's first as an NFL starting quarterback–but the two sides could not come to an agreement on salary or Hostetler's role.
	Hypothesis: The Steelers' biggest problem in bringing in a new backup quarterback might be salary.

Table 5: Negative Example

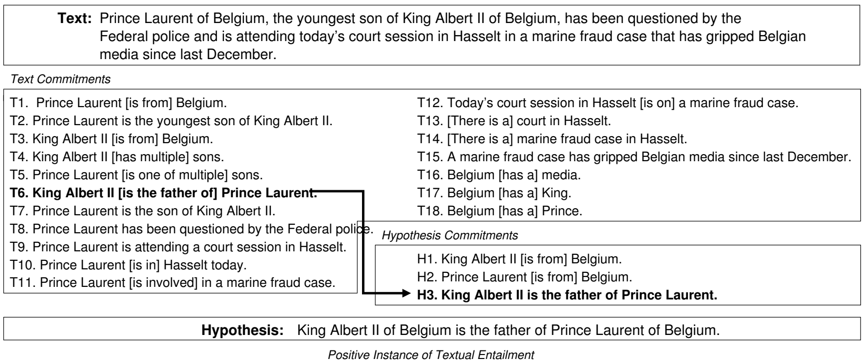
4. Generative Resource Creation Methods

In this section we describe how we experimented with methods for generating inference pairs which leverages the discourse commitment extraction framework introduced in (Hickl and Bensley, 2007) in order to generate candidate hypotheses from text passages retrieved from a document collection.

Under this approach, the *texts* included in the PASCAL RTE-1, RTE-2, and RTE-3 test sets (as well as those assembled using automatic methods for extracting positive and negative instances of TE) were analyzed using a *preprocessing* module that provides part-of-speech tagging,

4800 examples (2400 positive, 2400 negative) which can be used to train current RTE systems. The authors of (Harabagiu et al., 2006) created a collection of nearly 3000 (1500 positive, 1500 negative) instances of textual contradiction.

[4] Named entities were recognized using LCC's CICEROLITE named entity recognition system.



Text: Prince Laurent of Belgium, the youngest son of King Albert II of Belgium, has been questioned by the Federal police and is attending today's court session in Hasselt in a marine fraud case that has gripped Belgian media since last December.

Text Commitments

T1. Prince Laurent [is from] Belgium.
T2. Prince Laurent is the youngest son of King Albert II.
T3. King Albert II [is from] Belgium.
T4. King Albert II [has multiple] sons.
T5. Prince Laurent [is one of multiple] sons.
T6. King Albert II [is the father of] Prince Laurent.
T7. Prince Laurent is the son of King Albert II.
T8. Prince Laurent has been questioned by the Federal police.
T9. Prince Laurent is attending a court session in Hasselt.
T10. Prince Laurent [is in] Hasselt today.
T11. Prince Laurent [is involved] in a marine fraud case.
T12. Today's court session in Hasselt [is on] a marine fraud case.
T13. [There is a] court in Hasselt.
T14. [There is a] marine fraud case in Hasselt.
T15. A marine fraud case has gripped Belgian media since last December.
T16. Belgium [has a] media.
T17. Belgium [has a] King.
T18. Belgium [has a] Prince.

Hypothesis Commitments

H1. King Albert II [is from] Belgium.
H2. Prince Laurent [is from] Belgium.
H3. King Albert II is the father of Prince Laurent.

Hypothesis: King Albert II of Belgium is the father of Prince Laurent of Belgium.

Positive Instance of Textual Entailment

Figure 2: Commitments Extracted from a Positive Instance of TE.

named entity recognition, and syntactic dependency parsing. Keywords extracted and expanded from these preprocessed texts by methods described in (Hickl et al., 2007) were used to retrieve passages from the 2 GB AQUAINT-2 newswire collection. Passages were then ranked based on the density of keywords and submitted to a *sentence decomposition* module, which uses a set of heuristics to transform complex sentences containing subordination, relative clauses, lists, and coordination into sets of well-formed simple sentences. We then passed the passages to a *commitment extraction* module which used a series of extraction heuristics (described below) to enumerate some of the publicly-held beliefs – or *discourse commitments* – that could be inferred from a text passage.

We focused on generating the following five different types of commitments from retrieved passages.

Propositional Content: To capture assertions encoded by predicates and predicate nominals, we use semantic dependency information to generate ”simplified” commitments for each possible combination of their optional and obligatory arguments.

Supplemental Expressions: Rules to extract supplemental expressions, including appositives, *as*-clauses, parentheticals, non-restrictive relative clauses, and epithets were implemented in our weighted FST algorithm and used to create new sentences which specify the conventional implicature (CI) conveyed by the expression.

Relation Extraction: We used an in-house relation extraction system to recognize six types of semantic relations, including *artifact*, *general affiliation*, *organization affiliation*, *part-whole*, *social affiliation*, and *physical location*, from which we can build simple attribute commitments.

Coreference Resolution: We used an in-house coreference resolution (based on (Nicolae and Nicolae, 2006) module to resolve instances of pronominal and nominal coreference in order to expand the number of commitments available to the system.

Paraphrasing: A lightweight, knowledge-lean paraphrasing approach (Hickl et al., 2006) was used in order to expand the set of commitments considered by the system.

Each extracted commitment was then paired with its corresponding passage; commitments were presumed to be *hypotheses*, while passages were considered to be *texts*.

We also assembled inference pairs using extracted commitments and any of the PASCAL RTE *texts* that were used to retrieve passages from the corpus: commitments that received a sufficiently high similarity score with respect to a *text* were also assembled into an inference pair. (An example of an original text, a retrieved passage, and an extracted commitment are presented in Table 6.)

TE	Example
YES	**Text:** A Revenue Cutter, the ship was named for Harriet Lane, niece of President James Buchanan, who served as Buchanan's White House hostess.
	Passage: Named after the niece of President James Buchanan, the US Revenue Cutter Harriet Lane was a 750-ton side-wheel gunboat built in 1857.
	Commitment: A Revenue Cutter was named after the niece of President James Buchanan.

Table 6: Using Commitments to Assemble TE Pairs

We used a similar approach to generate training examples for an RTC system as well. After an h has been generated from a retrieved passage, we use the negation processing heuristics from (Harabagiu et al., 2006) to reverse the polarity of the predicate included in the h. (An example is provided in Table 7).

TC	Example
YES	**Text:** A Revenue Cutter, the ship was named for Harriet Lane, niece of President James Buchanan, who served as Buchanan's White House hostess.
	Passage: Named after the niece of President James Buchanan, the US Revenue Cutter Harriet Lane was a 750-ton side-wheel gunboat built in 1857.
	Commitment: A Revenue Cutter was [not] named after the niece of President James Buchanan.

Table 7: Using Commitments to Assemble TC Pairs

5. Evaluation

In this section, we present results from experiments which demonstrate the impact that automatically-created sources of training data can have on the end-to-end performance of state-of-the-art systems for recognizing TE and TC.

5.1. Validating Inference Pairs

After extracting an initial set of candidate TE and TC pairs an analysis was performed by validating samples with human annotators. For each of the techniques a random, fixed-size sampling was pulled from the generated pairs. A team of 8 annotators partitioned the data into three equal-sized sets of entailment pairs for evaluation. Each subset was examined by at least two annotators to judge the correctness of the generated pairs, and discrepancies were noted and resolved in conference by the annotators.

From this analysis we were able to identify and remove common sources of error using syntactic patterns. By combining lexical resources with our context-free pattern engine, we built rules to deal with the most error-prone language constructs in each of the construction methods. As shown in Table 8, this filtering process reduces the overall error rate by an average of 5.8% across the 4 methods.

Set	Error Before Filtering	Error After Filtering
Extraction Method 1	8.2%	5.3%
Extraction Method 2	5.8%	3.1%
Extraction Method 3	14.5%	6.4%
Generative Method	17.1%	8.3%

Table 8: Generation Errors Before and After Filtering.

Although desirable to achieve perfect error filtering on these training samples, we found most of the remaining sources of error are difficult issues of semantics and world knowledge that we have not yet been able to resolve. Even though imperfect filtering results in some slightly noisy training data, the resilience of our machine learning framework is able to overcome this discrepancy, such that there is a net benefit to the tasks of classifying TE and TC pairs.

5.2. Impact on Existing TE and TC Systems

Following the example of the PASCAL RTE evaluations, we evaluated the performance of our RTE and RTC systems along two dimensions: accuracy and average precision. We define *accuracy* as the percentage of inference pairs correctly classified by an RTE/RTC system. *Average precision*, is defined by (Glickman and Dagan, 2005) as:

$$\frac{1}{n} \times \sum_{i=1}^{n} \frac{\sum_{j=1}^{i}(correct_j)}{i} : correct_j \in 0, 1$$

This scoring metric assumes a sorted output based on *confidence weights*, with the highest confidence judgments appearing at the top of the sorted order.

Our first evaluation compared the impact of data generated using the various methods against a TE system trained just using the 2400-pair RTE development set. For the purposes of TE evaluation, data generated to provide contradiction pairs are considered negative entailment pairs.

Development Set	Accuracy	Average Precision
PASCAL Dev	0.6900	0.7152
Extraction Method 1	0.5428	0.5520
Extraction Method 2	0.5814	0.6186
Extraction Method 3	0.5303	0.5291
Generative Method	0.6649	0.6475

Table 9: Comparison of Training Corpora for Textual Entailment.

Since each extraction method is designed to create pairs of a specific polarity (e.g. method 1 generates only YES pairs), the methods acting in isolation actually perform worse than the baseline system trained only on the PASCAL Development Set. By taking combinations of data constructed from the various methods, however, we can achieve results comparable to a system trained on the PASCAL Development Set as demonstrated in Table 10.

Development Set	Accuracy	Average Precision
Methods 1 + 2	0.6246	0.6477
Methods 1 + 3	0.6194	0.6221
Methods 1 + 2 + 3	0.6550	0.6807
All Methods	0.6895	0.7019

Table 10: Combinations of Training Corpora Data.

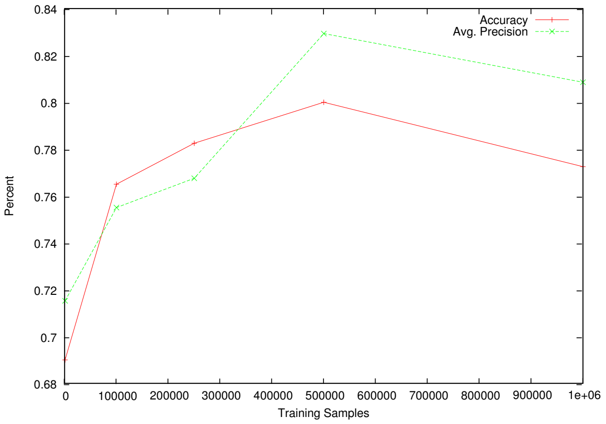


Figure 3: Entailment Scores with Increasing Training Data

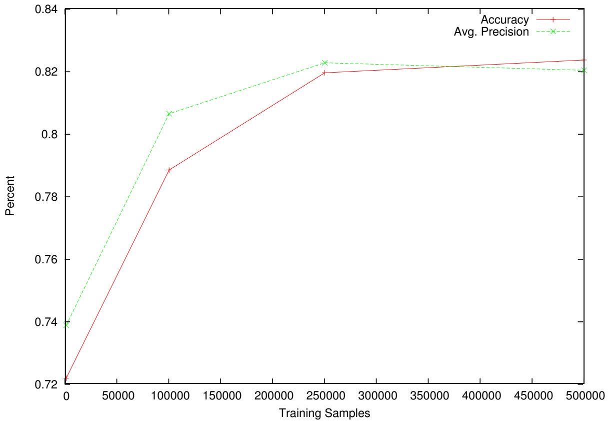


Figure 4: Contradiction Scores with Increasing Training Data

Having established that the automatically generated data compares favorably against the manually built development set for a fixed number of examples, we then examined the impact of increasing amounts of training data in the TE and TC systems. As shown in Figure 3 and Figure 4, larger amounts of training data provided significant performance boosts for both TE and TC classification, confirming the hypothesis suggested in (Hickl et al., 2006).

6. Conclusions

In this paper, we demonstrated how a battery of unsupervised techniques could be used in order to create large, high-quality corpora for textual inference applications, such as systems for recognizing textual entailment and recognizing textual contradiction. We described how more than 1 million pairs of training examples could be generated from the documents found in a 2 GB English newswire corpus.

In our experiments, we observed no significant difference in performance when equivalent number of hand-crafted or automatically-generated examples were used to train classifiers for recognizing textual entailment or textual contradiction. While we recognize that the techniques described in this paper may not provide the lexicosemantic or pragmatic knowledge needed by many textual inference applications, we expect that they can be exploited in order to provide the basic forms of linguistic knowledge needed to improve the performance of classification-based systems for computing textual inference.

Finally, our results confirm the hypothesis (first suggested in (Hickl et al., 2006)) that the performance of classification-based systems for RTE increases with the amount of available training data. In our experiments, increases in accuracy are observed when training on as many as 500,000 inference pairs; performance remains constant (or suffers slight degradation) with larger training corpora.

7. Acknowledgments

This material is based upon work funded in whole or in part by the U.S. Government. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of the U.S. Government.

8. References

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*.

P. Blackburn and J. Bos. 2005. *Representation and inference for natural language : a first course in computational semantics*. Center for the Study of Language and Information, Stanford, Calif.

Johan Bos and Katya Markert. 2006. When logical inference helps in determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL Recognizing Textual Entailment Conference*, Venice, Italy.

Chris Brockett and William Dolan. 2005. Support Vector Machines for Paraphrase Identification and Corpus Construction. In *Proceedings of the Third International Workshop on Paraphrasing*, Jeju, Korea.

John Burger and Lisa Ferro. 2005. Generating an Entailment Corpus from News Headlines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 49–54.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop*.

William Dolan and Chris Quirk. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of COLING 2004*, Geneva, Switzerland.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June. Association for Computational Linguistics.

Oren Glickman and Ido Dagan. 2005. A Probabilistic Setting and Lexical Co-occurrence Model for Textual Entailment. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor, USA.

Christine Gunlogson. 2001. *True to Form: Rising and Falling Declaratives as Questions in English*. Ph.D. thesis, University of California, Santa Cruz.

Aria Haghighi, Andrew Ng, and Christopher Manning. 2005. Robust textual inference via graph matching. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 387–394.

Sanda M. Harabagiu, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast and contradiction in text processing. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence (AAAI-2006)*.

Andrew Hickl and Jeremy Bensley. 2007. Recognizing Textual Entailment with LCC's Groundhog System. In *Proceedings of the Third PASCAL Challenges Workshop (to appear)*.

Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing Textual Entailment with LCC's Groundhog System. In *Proceedings of the Second PASCAL Challenges Workshop*.

Andrew Hickl, Kirk Roberts, Bryan Rink, Jeremy Bensely, Tobias Jungen, Ying Shi, and John Williams. 2007. Question Answering with LCC's Chaucer-2 at TREC 2007. In *Proceedings of the Sixteenth Text REtrieval Conference*.

R. Quinlan. 1998. C5.0: An Informal Tutorial. RuleQuest.

Rajat Raina, Andrew Y. Ng, and Chris Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*.

Robert Stalnaker, 1979. *Assertion*, volume 9, pages 315–332.

Ben Taskar, Simone Lacoste-Julien, and Michael Jordan. 2005a. Structured prediction via the extragradient method. In *Proceedings of Neural Information Processing Systems*.

Ben Taskar, Simone Lacoste-Julien, and Dan Klein. 2005b. A discriminative matching approach to word alignment. In *Proceedings of Human Language Technology Conference and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*.

Marta Tatu, Brandon Iles, John Slavick, Adrian Novischi, and Dan Moldovan. 2006. COGEX at the Second Recognizing Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop*.

Lucy Vanderwende, Arul Menezes, and Rion Snow. 2006. Microsoft Research at RTE-2: Syntactic Contributions in the Entailment Task: an implementation. In *Proceedings of the Second PASCAL Challenges Workshop*.