Recognizing Textual Entailment with LCC's GROUNDHOG System

Andrew Hickl, Jeremy Bensley, John Williams, Kirk Roberts, Bryan Rink, and Ying Shi
Language Computer Corporation
Richardson, Texas 75080 USA
groundhog@languagecomputer.com

Abstract

We introduce a new system for recognizing textual entailment (known as GROUNDHOG) which utilizes a classification-based approach to combine lexico-semantic information derived from text processing applications with a large collection of paraphrases acquired automatically from the WWW. Trained on 200,000 examples of textual entailment extracted from newswire corpora, our system managed to classify more than 75% of the pairs in the 2006 PASCAL RTE Test Set correctly.

1 Introduction

Processing textual entailment, or recognizing whether the information expressed in a *textual hypothesis* can be inferred from the information expressed in another *text*, can be performed in four ways. We can try to (1) derive linguistic information from the hypothesis-text pair, and cast the inference recognition as a classification problem; or (2) evaluate *the probability* that an entailment can exist between the hypothesis-text pair; (3) represent the knowledge from the hypothesis-text pair in some representation language that can be associated with an inferential mechanism; or (4) use the classical AI definition of entailment and build models of the world in which the hypothesis and the text are respectively true, and then check whether the models associated with the hypothesis are included in the models associated with the text. It is not clear which methodology is superior, but we argue that the first two methods rely more heavily on the accuracy and robustness of processing information from text, whereas the other two methods make use of reasoning technologies or model checking methods that apply to any kind of knowledge, not only to linguistic knowledge derived from text.

Although we believe that each of these methods should be investigated fully, we decided to focus only on the first method, which allows us to make use of some of the natural language processing tools developed at LCC. For this purpose, we have developed a system called GROUNDHOG, which relies on our ability to derive a variety of lexico-semantic information from text, including information about named entities, coreference, and syntactic and semantic dependencies. In addition, since textual paraphrases are a special case of entailment, we expect techniques used successfully in paraphrase recognition should also be useful for textual entailment. Even though the hypothesis often expresses less information than the text, it can generally be seen as a paraphrase of all or some portion of the text.

In our system, textual alignment is used to capture the candidate portions from the text and the hypothesis that could be paraphrases. Paraphrases for the candidate pair are acquired automatically from the World Wide Web. Features from the alignment process and from the acquired paraphrases are used together with semantic features and dependency features for training a classifier that decides the textual entailment. We claim that the quality of the classifier that we have trained is also due to the new sources of data that we have exploited.

The remainder of the paper is organized as follows. Section 2 describes the architecture of GROUNDHOG, while Section 3 details the linguistic processing that we have applied to each hypothesis-text pair. Section 4 describes the new sources of training data whereas Section 5 describes the lexical alignment methodology. Section 6 details the para-

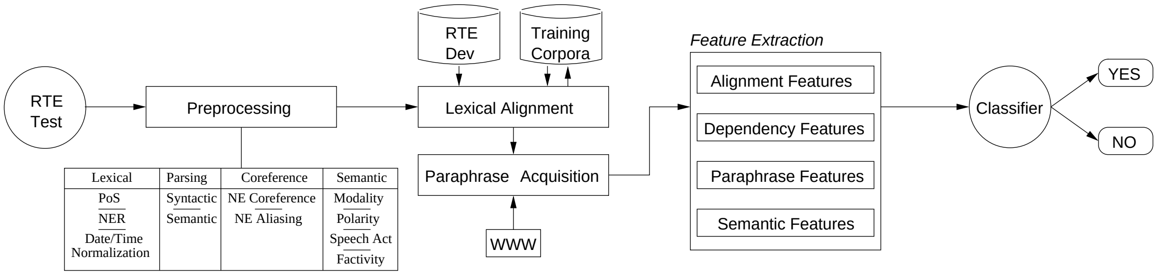

Figure 1: The GROUNDHOG System

phrase acquisition, and Section 7 provides details of the textual entailment classifier and Section 8 discusses evaluation results. Finally, Section 9 summarizes the conclusions.

2 The GROUNDHOG System

This section provides an overview of LCC's GROUNDHOG system for recognizing textual entailment that was evaluated in 2006 PASCAL RTE Challenge. In the rest of this section, we will present a brief overview of our system's pipeline (using Example 139 from the 2006 Test Set); details of each individual module are presented later in the paper.

Judgment	Paraphrase
YES	**Text:** The Bills now appear ready to hand the reins over to one of their two-top picks from a year ago in quarterback J.P. Losman, who missed most of last season with a broken leg.
	Hypothesis: The Bills plan to give the starting job to J.P. Losman.

Table 1: Example 139

In order to acquire the linguistic information needed to recognize textual entailment, text-hypothesis pairs are first sent to a *Text Preprocessing* module. Here, texts are syntactically parsed, semantic dependencies are identified using a semantic parser trained on predicate-argument annotations derived from PropBank (Palmer et al., 2005), entities are associated with named entity information from LCC's CiceroLite Named Entity Recognition system, time and space expressions are normalized to their ISO 9000 equivalents, coreferential entities are detected and resolved, and predicates are annotated with semantic features such as polarity and modality. Examples of the annotations performed for Example 139 presented in Table 2.

Once preprocessing is complete, text-hypothesis pairs are sent to a *Lexical Alignment* module which uses a Maximum Entropy-based classifier in order to determine the likelihood that either predicates or arguments selected from a text and a hypothesis lexically entail one another. Since performing this alignment requires access to large amounts of training data, this classifier is trained using two large corpora of positive and negative examples of textual entailment that we extracted from newswire corpora. Examples of predicates and arguments aligned by this module are presented below in Figure 2.

Named Entity Recognition
T: [The Bills]$_{\mathrm{team}}$ now appear ready to hand the reins over to one of their two-top picks from a year ago in quarterback [J.P. Losman]$_{\mathrm{person}}$, who missed most of last season with a broken leg.
H: [The Bills]$_{\mathrm{team}}$ plan to give the starting job to [J.P. Losman]$_{\mathrm{person}}$.
Semantic Parsing
T: [The Bills]$_{\mathrm{Arg0}}$ [now]$_{\mathrm{ArgM-tmp}}$ appear ready to [hand]$_{\mathrm{predicate}}$ [the reins]$_{\mathrm{Arg1}}$ over to [one of their two-top picks]$_{\mathrm{Arg2}}$ from a year ago in quarterback J.P. Losman, who missed most of last season with a broken leg.
H: [The Bills]$_{\mathrm{Arg0}}$ plan to [give]$_{\mathrm{predicate}}$ [the starting job]$_{\mathrm{Arg1}}$ to [J.P. Losman]$_{\mathrm{Arg2}}$.
Coreference/Aliasing
T: [The Bills]$_{\mathrm{i}}$ now appear ready to hand the reins over to [one of their two-top picks]$_{\mathrm{j}}$ from a year ago in [quarterback]$_{\mathrm{j}}$ [J.P. Losman]$_{\mathrm{j}}$, who missed most of last season with a broken leg.
H: [The Bills]$_{\mathrm{i}}$ plan to give the starting job to [J.P. Losman]$_{\mathrm{j}}$.
Temporal/Spatial Normalization
T: The Bills [now]$_{2006-01-01}$ appear ready to hand the reins over to one of their two-top picks from [a year ago]$_{2005-01-01}$ in quarterback J.P. Losman, who missed most of last season with a broken leg.
H: The Bills plan to give the starting job to J.P. Losman.

Table 2: Annotated Example

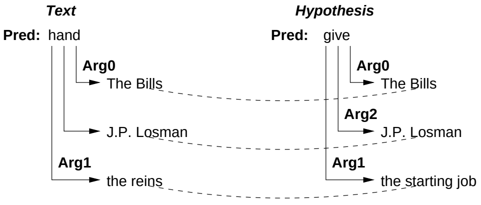

Figure 2: Alignment Graph

The most likely pairs of aligned constituents identified by the Lexical Alignment module are then sent to a *Paraphrase Acquisition* module, which extracts paraphrases that contain pairs of aligned constituents from the WWW. The top 8 sentences containing the aligned terms from Figure 2 are presented in Table 3. Since not all acquired paraphrases will be synonymous with either the text or hypothesis, a complete-link clustering algorithm (similar to (Barzilay and Lee, 2003)) was used to cluster paraphrases into sets that are presumed that convey the same content.

Judgment	Paraphrase
YES	*The Bills* have gone with quarterback *J.P. Losman*
YES	*The Bills* decided to put their trust in *J.P. Losman*
YES	*The Bills* benched Bledsoe in favor of *Losman*
NO	*Buffalo* gave away to acquire *J.P. Losman*
YES	*Buffalo* is now molding their quarterback-of-the-future *J.P. Losman*
YES	*Buffalo* coach Mike Mularkey will start *J.P. Losman*
YES	*The Bills* have turned over the keys to the offense to *J.P. Losman*
NO	*The Buffalo Bills'* 2005 season hinges on the performance of quarterback *J.P. Losman*

Table 3: Phrase-Level Alternations

Semantic information is then combined into an *Entailment Classifier* which determines the likelihood that a textual entailment relationship exists for a particular pair of texts. Four classes of features are extracted: (1) *alignment features*, which compare properties of aligned constituents, (2) *dependency features*, which compare entities and predicates using dependencies identified by a semantic parser, (3) *paraphrase features*, which determine whether passages extracted from the text and hypothesis match acquired paraphrases, and (4) *semantic features*, which contrast semantic values assigned to predicates in each example sentence. Based on these features, the Entailment Classifier outputs both an entailment classification (either `YES` or `NO`) and a confidence value, which is used to rank examples for the final RTE submission.

3 Preprocessing

GROUNDHOG annotates each pair of examples in its training and testing corpora with four types of information:

Lexical Information. After tokenization, sentences are sent to LCC's CiceroLite named entity recognition (NER) system to be associated with one of more than 150 different named entity classes. Temporal expressions, (including dates and times) and spatial expressions (including names of most political and geographic locations) are then then sent to LCC's TASER temporal and spatial normalization system (Lehmann et al., 2005), which maps these expressions to their ISO 9000 equivalents.

Syntactic and Semantic Parse Information. Sentences are then sent to GROUNDHOG's syntactic and semantic parsers. Syntactic parsing is performed using LCC's own implementation of the Collins Parser (Collins, 1996), while semantic parsing is performed using a Maximum Entropy-based semantic role labeling system trained on the predicate-argument annotations found in PropBank (Palmer et al., 2005).

Coreference Information. We used a combination of heuristics and lexica from LCC's CiceroLite to identify coreferential named entities and to perform name aliasing for all of the entities found in each text-hypothesis pair.

Semantic Information. We also employed a set of heuristics to annotate text passages with semantic features. First, predicates are annotated with *polarity* information: verbs and event-denoting nominals within the syntactic scope of a overt negative marker (e.g. *not*, *n't*, *never*) or a negation-denoting verb (e.g. *deny, refuse*) are assigned a `false` value. A second set of heuristics is also used to identify cases where a truth value for a proposition cannot be determined: predicates occurring in conditional constructions or with modal auxiliaries, speech act verbs, or *psych*-verbs are marked as `unresolved`.

4 Creating New Sources of Training Data

In order to augment the training data provided by the Challenge organizers, we experimented with techniques to extract positive and negative examples of textual entailment from large newswire corpora. We gathered more than 200,000 additional entailment pairs that we used to train GROUNDHOG.

Positive Examples. Following an idea proposed in (Burger and Ferro, 2005), we created a corpus of approximately 101,000 textual entailment examples by pairing the headline and first sentence from newswire documents. Since the headlines and first sentences of newswire texts are often used to synopsize the content of a document, we found that most extracted pairs could be considered to be positive examples of textual entailment. In order to increase the likelihood of including only positive examples, pairs were filtered that did not share an entity (or an NP) in common between the headline and the first sentence, as were pairs that discussed sports results or stock prices. In a sample of 2500 pairs selected at random, 2296 (91.8%) were judged by human annotators as positive.

Judgment	Example
YES	**Text:** Sydney newspapers made a secret deal not to report on the fawning and spending during the city's successful bid for the 2000 Olympics, former Olympics Minister Bruce Baird said today.
	Hypothesis: Papers Said To Protect Sydney Bid
YES	**Text:** An IOC member expelled in the Olympic bribery scandal was consistently drunk as he checked out Stockholm's bid for the 2004 Games and got so offensive that he was thrown out of a dinner party, Swedish officials said.
	Hypothesis: Officials Say IOC Member Was Drunk

Table 4: Positive Examples

Negative Examples. Two approaches were used to gather negative examples for our training set. First, we extracted pairs of sequential sentences that included mentions of the same named entity from a large newswire corpus. We gathered more than 98,000 examples of this type from nearly 700,000 documents; human annotators deemed 97.5% (2438/2500) of these examples to be negative examples. In order to gather more negative examples, we extracted approximately 21,000 pairs of sentences linked by discourse connectives such as *even though*, *although*, *otherwise*, *in contrast* and *but*. In an analysis of 1000 of these examples, annotators judged 942 (94.2%) to be negative for textual entailment.

Judgment	Example
NO	**Text:** One player losing a close friend is Japanese pitcher Hideki Irabu, who was befriended by Wells during spring training last year.
	Hypothesis: Irabu said he would take Wells out to dinner when the Yankees visit Toronto.
NO	**Text:** According to the professor, present methods of cleaning up oil slicks are extremely costly and are never completely efficient.
	Hypothesis: *In contrast*, he stressed, Clean Mag has a 100 percent pollution retrieval rate, is low cost and can be recycled.

Table 5: Negative Examples

5 Lexical Alignment

Several previous approaches to the RTE task have argued that term-based measures can be used to successfully identify instances of lexical entailment in texts. While these approaches performed reasonably well in the 2005 PASCAL RTE Challenge, we feel that even better results could be obtained by combining these linguistically-naive probabilistic approaches with richer forms of lexico-semantic information. In this section, we show that by using a machine learning-based classifier which combines lexico-semantic information from a wide range of sources, we are able to accurately identify lexical entailment relationships with over 90% accuracy.

We believe the identification of lexical entailment can be cast as a classification problem which uses lexico-semantic features in order to compute an alignment probability *p(a)*, which corresponds to the likelihood that a term selected from a text entails a term selected from a corresponding hypothesis. We used constituency information from a chunk parser to decompose texts and hypotheses into a set of disjoint segments known as "alignable chunks". Alignable chunks from the text (C_t) and the hypothesis (C_h) are then assembled into an alignment matrix ($C_t \times C_h$). Each pair of chunks ($p \in C_t \times C_h$) is then submitted to a Maximum Entropy-based classifier which determines whether or not the pair of chunks represents a case of lexical entailment.

Three classes of features were used in the Alignment Classifier: (1) a set of statistical features (including cosine similarity, and (Glickman and Dagan, 2005)'s lexical entailment probability), (2) a set of lexico-semantic features (including WordNet Similarity (Pedersen et al., 2004), named entity class equality, and part-of-speech equality), and (3) a set of string-based features (such as Levenshtein edit distance and morphological stem equality).

We used a two-step approach to obtain sufficient training data for the Alignment Classifier. A total of 10,000 alignment pairs extracted from the 2006 Development Set were annotated by humans as either positive or negative examples of lexical entailment. These annotations were used to train a hillclimber that was used to annotate a larger set of 450,000 alignment pairs selected at random from the training corpora described in Section 4 that was then used to train a Maximum Entropy-based classifier that was used on the 2006 Test Set. Table 6 presents results from GROUNDHOG's hillclimber- and Maximum Entropy-based Alignment Classifiers on a sample of 1000 alignment pairs selected at random from the 2006 Test Set.

Classifier	**Training Set**	**Precision**	**Recall**	**F-Measure**
Hillclimber	10K pairs	0.837	0.774	0.804
Maximum Entropy	10K pairs	0.881	0.851	0.866
Maximum Entropy	450K pairs	0.902	0.944	0.922

Table 6: Performance of Alignment Classifier

6 Paraphrase Acquisition

Much recent work on automatic paraphrasing (Dolan et al., 2004; Barzilay and Lee, 2003; Shinyama et al., 2002) has used relatively simple statistical techniques to identify text passages that contain the same information from parallel corpora. Since sentence-level paraphrases are generally assumed to contain information about the same event, these approaches have generally assumed that that all of the available paraphrases for a given sentence will include at least one pair of entities which can be used to extract sets of paraphrases from text.

GROUNDHOG uses a similar approach to gather phrase-level alternations for each entailment pair. In our system, the two highest-confidence entity alignments returned by the Lexical Alignment module was used to construct a query which was used to retrieve the top 500 documents from *Google*, as well as all matching instances from the LCC training corpora described in Section 4. This method did not always extract patterns that were true paraphrases of either the text or the hypothesis. In order increase the likelihood that only true paraphrases were considered as phrase-level alternations for an example, extracted sentences were clustered using complete-link clustering using a technique proposed in (Barzilay and Lee, 2003).

7 Entailment Classifier

GROUNDHOG uses an Entailment Classifier in order to determine whether an entailment relationship exists for a text-hypothesis pair. After experiment with machine learning techniques (including Maximum Entropy and Support Vector Machines), we found that decision trees (as implemented in C5.0 (Quinlan, 2003)) performed best on the 2006 Test Set. [1] Confidence values returned by the C.5 classifier were normalized and used to rank examples for the final submission. In the rest of this section, we describe the four different types of features used in the Entailment Classifier.

7.1 Alignment Features

We used three features based on lexical alignment: (1) a *longest common string* feature which computed the longest contiguous string common to both the text and hypothesis, (2) a *unaligned chunk* feature equal to the number of chunks in the hypothesis not aligned with a chunk from the text, and (3) a feature based on (Glickman and Dagan, 2005)'s *lexical entailment probability* which was calculated using frequency counts returned by the *Google* API.

7.2 Dependency Features

Four features were computed from the PropBank-style annotations assigned by the semantic parser: (1) a boolean *entity arg-match* feature which fired when aligned entities were assigned the same argument role label, (2) a boolean *entity near arg-match* feature which collapsed Arg_1 and Arg_2 (as well as the Arg_M subtypes) into single categories for the purpose of counting matches, and finally, (3) a *predicate arg-match* feature and (4) a *predicate near arg-match* feature which compared the role labels associated with aligned predicates.

7.3 Paraphrase Features

Three types of features were derived from the paraphrases acquired for each pair: (1) a *single pattern match* feature which fired when a paraphrase matched either the text or the hypothesis, (2) a *both pattern match* feature which fired when paraphrases matched both the text and the hypothesis, and (3) a *category match* feature which fired when paraphrases could be found from the same paraphrase cluster that matched both the text and the hypothesis.

7.4 Semantic Features

Two features were used that took advantage of the truth values that the Preprocessor assigned to predicates: (1) a boolean *truth-value mismatch* feature which fired when two aligned predicates differed in any truth value, and (2) a boolean *polarity mismatch* feature which fired when predicates were assigned opposite polarity values.

8 Evaluation

Our results for the 2006 RTE Challenge task are summarized in Table 7 below.

Task	Accuracy	Average Precision
QA-test	0.6950	0.8237
IE-test	0.7300	0.8351
IR-test	0.7450	0.7774
SUM-test	0.8450	0.8343
Total	**0.7538**	**0.8082**

Table 7: Performance on the 2006 Test Set

We found that GROUNDHOG's performance differed markedly across the 4 Challenge subtasks.

[1] We used attribute winnowing and a pruning confidence level of 5% in our final model.

While the system was able to correctly identify entailment in nearly 85% of the examples from the Multi-Document Summarization (SUM) subtask, performance dipped below 70% on the Question Answering (QA) set.

Figure 3 presents results that describe the impact of the four feature types used in GROUNDHOG's Entailment Classifier.

		+ Align	+ Dep	+Para
Semantic	58.00	66.25	71.25	75.38
Paraphrase	65.88	69.13	73.62	
Dependency	62.50	68.00		
Alignment	65.25			

Figure 3: Feature Performance

While the best results were obtained by combining all four sets of features (75.38%), the largest gains in accuracy were obtained by incorporating features based on paraphrases extracted by the Paraphrase Acquisition module. When used alone, paraphrase features performed well, making the correct entailment classification nearly 66% of the time.

We found that access to large amounts of training data significantly impacted our overall score. When trained only on the 800 text-hypothesis pairs in the PASCAL 2006 Development Set (Dev), GROUNDHOG correctly classified 65.25% of the examples in the Test Set;when the classifier was trained on the more than 200,000 examples in LCC's training corpora, performance was increased by 10%.

Training Set	Accuracy	Precision	Δ Accuracy
Pascal 2006 Dev	65.25%	0.5509	n/a
25% LCC Corpora + Dev	67.00%	0.7333	+1.75%
50% LCC Corpora + Dev	72.25%	0.7446	+5.25%
75% LCC Corpora + Dev	74.38%	0.7992	+2.13%
100% LCC Corpora + Dev	75.38%	0.8082	+1.00%

Table 8: Impact of Training Data

9 Conclusion

We have described our methodology for recognizing textual entailment that was utilized in the 2006 PASCAL RTE Challenge. We are satisfied with our results from this evaluation, which we feel illustrates the potential of one of the four textual entailment frameworks that we considered in Section 1.

10 Acknowledgments

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

References

Regina Barzilay and Lillian Lee. 2003. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. *Proceedings of HLT-NAACL*, pages 16–23.

John Burger and Lisa Ferro. Generating an Entailment Corpus from News Headlines. 2005. *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 49–54. Association for Computational Linguistics, Ann Arbor, Michigan.

Michael Collins. 1996. A New Statistical Parser Based on Bigram Lexical Dependencies. *Proceedings of the 34th Annual Meeting of the ACL*, Santa Cruz.

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

Oren Glickman and Ido Dagan 2005. A Probabilistic Setting and Lexical Co-occurrence Model for Textual Entailment. *ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 30 June 2005, Ann Arbor, USA.

John Lehmann, Paul Aarseth, Luke Nezda, Murat Deligonul, and Andrew Hickl. 2005. TASER: A Temporal and Spatial Expression Recognition and Normalization System. *Proceedings of the 2005 Automatic Content Extraction Conference. Gaithersburg, MD.*

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics* **31**:1.

T. Pedersen, S. Patwardhan, and J. Michelizzi 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. *Proceedings of the Nineteenth National Conference on Artificial Intelligence* (AAAI-04), July 25-29, 2004, San Jose, CA (Intelligent Systems Demonstration).

Quinlan, J.R. 2003. C5.0 Online Tutorial. *http://www.rulequest.com*.

Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. 2002. Automatic Paraphrase Acquisition from News Articles. *Proceedings of Human Language Technology Conference*, San Diego, USA.