# Negation, Contrast and Contradiction in Text Processing

**Sanda Harabagiu, Andrew Hickl and Finley Lacatusu**
Language Computer Corporation
Richardson, TX 75080
{sanda, andy, finley}@languagecomputer.com

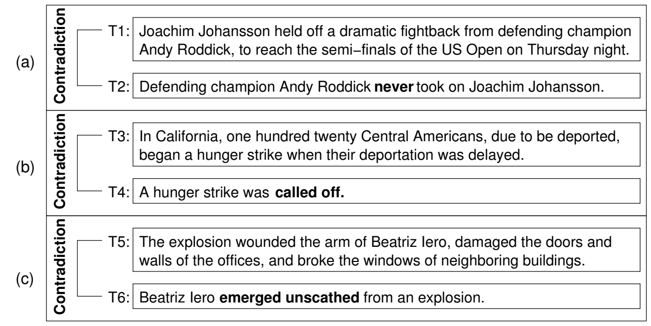
**Abstract**

This paper describes a framework for recognizing contradictions between multiple text sources by relying on three forms of linguistic information: (a) negation; (b) antonymy; and (c) semantic and pragmatic information associated with the discourse relations. Two views of contradictions are considered, in which a novel method of recognizing contrast and of finding antonymies are described. Contradictions are used for informing fusion operators in question answering. Our experiments show promising results for the detection of contradictions.

## 1. Introduction

Contradictions occur whenever information that is communicated in two different texts is incompatible. Incompatibilities are manifested in many ways. We have focused on contradictions that originate when using (i) negation; (ii) antonymy; or (iii) semantic and pragmatic information that is characteristic of CONTRAST discourse relations. Figure 1(a) illustrates an example of contradictions that arise from the usage of the negative "*never*". Both texts use predicates that have similar meanings, employ the same arguments, but text $T_2$ inverses the truth value through negation. Figure 1(b) illustrates an example of contradiction marked by the usage of the antonyms "*begun*" and "*call off*" predicated over the same arguments. Figure 1(c) exemplifies a contradiction in which the facts inferred from text $T_5$ (e.g "Beatriz Iero was wounded by the explosion") are denied by the facts inferred from text $T_6$ (e.g. "Beatriz Iero was not wounded by the explosion" ). The semantic and pragmatic processes that allow such inferences are typical for the CONTRAST discourse relations.

Contradictions need to be recognized by Question Answering (Q/A) systems or by Multi-Document Summarization (MDS) systems. The recognition of contradictions is useful to fusion operators, that consider information originating in different texts. When compatible and non-redundant information is discovered in different sources, fusion operators select text snippets from all sources for producing informative answers. However, when contradictory information is discovered, the answer selects information



T1: Joachim Johansson held off a dramatic fightback from defending champion Andy Roddick, to reach the semi–finals of the US Open on Thursday night.
T2: Defending champion Andy Roddick **never** took on Joachim Johansson.

T3: In California, one hundred twenty Central Americans, due to be deported, began a hunger strike when their deportation was delayed.
T4: A hunger strike was **called off.**

T5: The explosion wounded the arm of Beatriz Iero, damaged the doors and walls of the offices, and broke the windows of neighboring buildings.
T6: Beatriz Iero **emerged unscathed** from an explosion.

Figure 1: Contradictions.

from only one of the texts, discarding its contradiction. For example, Figure 2 presents a question and set of two contradictory answers. In a case like this one, answer fusion operators must identify that the sentences in $\text{Answer}_1$ and $\text{Answer}_2$ represent contradictions that needs to be resolved in some way, either by intervention from the user of the Q/A system or through additional sources of feedback obtained from knowledge bases or additional answers extracted from the text collection. Here, recognizing that $\text{Answer}_2$ contradicted $\text{Answer}_1$ could provide a Q/A system with evidence that could be used to filter spurious answers automatically.

| |
|---|
| **Question**: When did Pakistan test its Shaheen-2 ballistic missile? |
| $\textbf{Answer}_1$: The source noted that the Shaheen-2, with a range of 2400 km, has **never** been tested by Pakistan. |
| $\textbf{Answer}_2$: Pakistan has said that it performed several tests of its 2300 km-range Shaheen-2 missile **in September 2004**. |

Figure 2: Contradictory Answers.

The main contribution of this paper is a novel and dual framework for recognizing contradictions. This framework combines the processing and removal of negation, the derivation of antonymy with the detection of CONTRAST relations. Antonymy is discovered by mining WordNet paths that extend an encoded antonymy. These paths are also used for recognizing CONTRAST relations, as they belong to six features designed specifically for capturing *opposing information*. Another novelty of this paper is the casting of the recognition of the CONTRAST discourse relation as a classification problem that operates on the results of textual alignment. Textual alignment has been used previously for recognizing paraphrases, and more recently

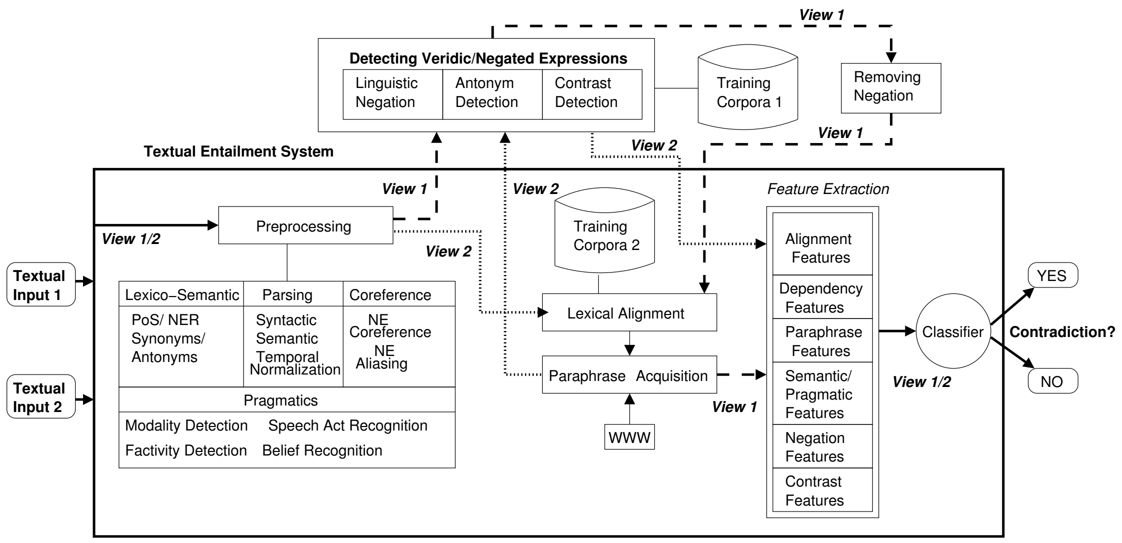


Figure 3: The Architecture Used for Recognizing Contradictions with the Help of Textual Entailment.

for detecting textual entailment (Glickman & Dagan 2005; Haghighi, Ng, & Manning 2005).

The remainder of this paper is organized as follows. Section 2 gives an overview of our method for recognizing contradictions. Section 3 details the method used for processing negations in texts and questions. Section 4 describes the process of discovering CONTRAST relations, whereas Section 5 gives a detailed description of the evaluation results. Section 6 summarizes the conclusions.

## 2. Recognizing Contradictions

Textual contradictions represent a form of textual inference. Another form of textual inference that has received a lot of attention in the recent years is *textual entailment*. To our knowledge, textual contradictions have not been studied as actively in the recent past, although both forms of inference are needed by Q/A systems. Both contradictions and entailments consider two textual inputs (sentences, paragraphs or questions) to deduce that one text contradicts the other, or one text can be entailed from the other, respectively.

In logic, contradictions arise whenever we find a simultaneous assertion of a proposition and its negation. Given the relationship between entailment and contradiction used in resolution, we have contemplated the usage of textual entailment methods for the recognition of textual contradictions. Our textual entailment system is based on a method that derives linguistic information from the two text inputs and casts the entailment recognition as a classification problem. In using textual entailment for discovering textual contradictions, we have considered two different views:

◇ View 1: Contradictions are recognized by identifying and removing negations of propositions and then testing for textual entailment;

◇ View 2: Contradictions are recognized by deriving linguistic information from the text inputs, including information that identifies (a) negations, (b) contrasts, or (c) oppositions (antonyms) and by training a classifier based on examples, similarly the way a classifier for entailment is trained.

Our framework for discovering textual contradictions in based on the architecture illustrated in Figure 3. The processing flow corresponding to View 1 is represented by a dashed line, the processing flow corresponding to View 2 is represented by a dotted line, whereas a normal line indicates an overlap of the two processing flows.

Our textual entailment system consists of (a) a PREPROCESSING MODULE, which derives linguistic knowledge from the text pair; (b) an ALIGNMENT MODULE, which takes advantage of the notions of lexical alignment and textual paraphrases; and (c) a CLASSIFICATION MODULE, which contains a feature extractor and a classifier. We used modules from the textual entailment system in the two processing flows that correspond to either of our views on recognizing contradictions. In addition, we have used a module that enables us to detect veridic and negated expressions in the textual inputs as well as a module that eliminates the detected negations.

The preprocessing module annotates both text inputs with four forms of linguistic information: (1) lexico-semantic information; (2) parsing information; (3) coreference information; and (4) pragmatic information. The lexico-semantic information consists of (i) part-of-speech information; (ii) synonymy and antonymy information derived from WordNet (Fellbaum 1998); and (iii) named entity classes provided by our named entity recognition (NER) system capable of distinguishing more than 150 different classes. The parsing information consists of syntactic and semantic parses. The syntactic parse is generated by Collins' parser (Collins 1999), whereas the semantic parse is produced by our parser (Surdeanu *et al.* 2003) trained on the annotations from PropBank (Palmer, Gildea, & Kingsbury 2005) and NomBank (Meyers *et al.* 2004). The parses contain also temporal and spatial normalizations of temporal expressions (including dates and time intervals) and spatial expressions (including names of politic and geographic locations). The coreference information is based on (a) the recognition of the name aliases; and (b) the recognition of named entity coreference. The pragmatic linguistic information that is derived is based on the recognition of modal auxiliaries (e.g. "*would*", "*could*"), factive verbs (e.g. "*manage*"), belief verbs (e.g. "*consider*", "*believe*"), or lexicons associated with speech acts (e.g. "*say*", "*announced*"). After preprocessing the text units, the process flow associated with View 1 leads to the detection of veridic and negated expressions, whereas the process flow associated with view 2 leads to the alignment module.

Veridicity and negation are recognized by the techniques described in Section 3 of this paper. The contrasts are detected based on a large training corpus that combines our annotations of contrasts and contradictions with a corpus generated with the method described in (Marcu & Echihabi 2002). We represent this combined corpora as Training Corpora 1 in Figure 3. When linguistic negation is discovered, it is removed with techniques detailed in Section 3 of this paper. Negation removal is performed only in the processing corresponding to View 1.

Both processing flows use the alignment module of the textual entailment system (only View 1 needs to detect and remove negated expressions before aligning the texts). The alignment module, designed originally for textual entailment, is based on the assumption that a text $T_1$ that is entailed from a text $T_2$ can also be seen as a paraphrase of text $T_2$. Therefore, textual alignment is used for capturing the candidate portions from the two texts that could be paraphrases. The alignment module contains the lexical alignment and the paraphrase acquisition components. The paraphrase acquisition component is described in (Hickl *et al.* 2006).

As described in (Hickl *et al.* 2006), lexical alignment was performed using a Maximum Entropy-based classifier trained on a total of 450,000 alignment pairs extracted from the PASCAL RTE 2006 development set[1] and a collection of more than 200,000 examples of textual entailment mined from newswire corpora and the world wide web.

In the second view of recognizing contradictions, the process flows to the detection of veridic/negated expressions, which was described previously. This module enables the detection of negation and contrast features. Then both processing flows complete the feature extraction that also depends on dependency features and semantic/pragmatic features derived from the knowledge produced by the preprocessing module. All these features (illustrated in Figure 5) are used by the inference classifier, which is trained on the Training Corpora 1 (based on examples of contradictions). After experimenting with several machine learning techniques (including Maximum Entropy and Support Vector Machines), we found that decision trees (as implemented in C5.0 (Quinlan 1998)) performed best when trying a 10-fold cross-validation of the examples of contradictions. We have found similar results inferring textual entailment. In both cases confidence results returned by the C5.0 classifier were normalized and used to rank the test example decisions.

[1]The PASCAL Recognizing Textual Entailment (RTE) Challenge evaluates the recognition of the inference of the meaning of one text fragment from another text. Further information can be found at: http://www.pascal-network.org/Challenges/RTE

## 3. Processing Negation in Text

Even though negation is considered to be a universal property of all human languages (Greenberg 1978), the processing (and interpretation) of negation in natural language texts has long remained an open question. While some recent work, reported in (Chapman *et al.* 2001), has focused on the detection of overtly-marked negation in texts, we know of no current work in NLP which has taken steps towards the interpretation of negated constituents in text.

We process *negation* by considering (i) overt (directly licensed) negation, and (ii) indirectly licensed negation. In English, direct licensors of negation include (1) overt negative markers such as the bound morpheme *n't* (e.g. "*don't*", "*can't*", "*won't*") and its free counterpart *not*, (2) negative quantifiers like *no* (including expressions such as "*no one*" and "*nothing*"), and (3) strong negative adverbs like "*never*". Examples of indirectly licensed negations include: (1) verbs or phrasal verbs (e.g. "*deny*", "*fail*", "*refuse*", "*keep from*"); (2) prepositions (e.g. "*without*", "*except*"), (3) weak quantifiers (e.g. "*few*", "*any*", "*some*"), and (4) traditional negative polarity items such as "*a red cent*" or "*any more*".

### 3.1 Detecting Negation in Text

In this section, we describe how we detect three types of negated constituents in texts: (1) negated events, (2) negated entities, and (3) negated states. We also describe our method for reversing the polarity of negated events, entities, and states in texts. To detect negations we use the following steps:

◇ STEP 1. Texts are preprocessed, as illustrated in Figure 3, and overt and implicit markers of negation (listed in a large lexicon of possible negation-denoting terms) are flagged in individual sentences.

◇ STEP 2. (Detect negated events) We detect instances of events based on the results of the semantic parser used in the preprocessing of texts. We filter out all events that are not having predicates marked as negated by an overt or implicit marker. The scope of the marker is the entire predicate-argument structure. We detect instances of events using two separate machine learning-based event detection systems; our verbal event detection system is trained using annotations derived form PropBank (Palmer, Gildea, & Kingsbury 2005), while our nominal event detection system is trained using features from NomBank (Meyers *et al.* 2004). Events detected by these systems are marked as negated if they fall within the syntactic scope of an overt or implicit negative marker. For example, in a sentence like "*The source noted that the Shaheen-2 had never been tested by Pakistan*", the predicate-argument structure headed by the verb "*tested*" is annotated with a `false` truth value.

◇ STEP 3. (Detect negated entities) We consider a negated entity to be any noun phrase that falls within the scope of an overt negative quantifier (e.g. *no*) or a non-veridical quantifier such as *few*, *some*, or *many*. Unlike negated events, where we assume the scope of a negative marker is assumed to encompass an entire VP, we restrict the scope of negative quantifiers to the widest scope interpretation of the modified NP. For example, in sentence "*No items in the museum's*

*collection of Japanese artifacts were obtained illegally.*", we consider the entire NP "*items in the museum's collection of Japanese artifacts*" to be considered the scope of negation.
◇ STEP 4. (Detect negated states) In order to detect instances of states in texts, we constructed an ontology of state-denoting terms from WordNet. We began by selecting approximately 175 state-denoting hypernyms (e.g. *symptom*, *condition*, or *situation*); nominals that were listed as hyponyms of these terms were marked as potentially state-denoting. We then used these terms to train a Maximum Entropy-based classifier in order to identify other state-denoting terms in WordNet. As with negated entities, we consider a negated state to be equal to the widest scope interpretation of a state-denoting noun phrase that falls within the scope of a negative marker.

In addition to detecting negation, we have also implemented techniques for reversing the polarity of negated events, entities, and states in texts. While removing overt negative markers (e.g. *no*, *not*, *never*) is relatively simple solution to this problem for many cases, this approach does not account for examples featuring negative-denoting verbs like *deny* or adverbs such as *hardly*.

In order to reverse the polarity of these examples, we have used a multi-tiered approach that utilizes (1) antonyms extracted from WordNet (e.g. *deny*:*admit*, *refuse*:*permit*), (2) antonyms identified using the automatic techniques described in the next Section , and (3) paraphrases derived as part of the system described in (Hickl *et al.* 2006).

## 3.2 Answering Negative Questions

Although today's state-of-the-art question-answering (Q/A) systems are beginning to incorporate techniques of the answering of semantically complex questions, we know of no current Q/A system that is capable of answering questions that contain overt or implicit forms of negation.

As can be seen in Table 1, questions that contain negated elements seek sets of entities or events which correspond to the semantic category of an expected answer type and satisfy the negated form of the proposition expressed by a question. For example, in a question like "*What countries have not won the World Cup?*", an appropriate answer includes the set of all countries for which the proposition "*won the World Cup*" is `false`.

| | |
|---|---|
| Q1 | What countries have **not** won the World Cup? |
| Q2 | Which Indonesian islands are **not** in the Southern Hemisphere? |
| Q3 | What US Army vehicles are **not** personnel transports? |
| Q4 | What nerve agents does Russia have **other than** VX gas? |

Table 1: Negative Questions

We have experimented with three techniques for answering questions like the ones in Table 1. The techniques are:
○ TECHNIQUE 1. The first technique relies on the detection of negation. Answers to negative questions were then returned from the set of candidate answers that were found in contexts where the proposition found in the question was deemed to be false. Even though this approach returned high-precision answers to some negative questions, this strategy's recall was ultimately limited by the number of candidate answer passages found in a text collection that were negated.
○ TECHNIQUE 2. In the second technique, in order to increase recall, we adapted a strategy first proposed by (Bierner & Webber 2000) for the answering of so-called *exceptive questions* like (Q4) above. Under this approach, we transformed negative questions like "*What Indonesian islands are not found in the Southern Hemisphere?*" into two corresponding questions (1) a positive analog of the original negative question (e.g. "*What Indonesian islands are found in the Southern Hemisphere?*") and (2) a second question that sought the full extension of the original question's answer type term (e.g. "*What Indonesian islands are there?*"). Answers to the original negative question were then derived by returning the set of entities that were not included in the intersection of (1) and (2).
○ TECHNIQUE 3. In the third technique, we utilized the system for recognizing textual entailment (RTE) described in (Hickl *et al.* 2006) in order to find candidate answers which entailed (or at least closely approximated) the semantic content of the proposition expressed by a negative question. For this approach, we used the keyword density algorithm described in (Harabagiu *et al.* 2003) in order to retrieve a set of candidate answer passages which featured keywords extracted from the negative question. Negative questions were also transformed into declarative statements using a set of simple heuristics which reordered the inverted subject and verb and replaced the question's WH phrase with a noun phrase corresponding to the EAT of the question.) Each candidate answer passage was then submitted to the RTE system; answers that were deemed to be entailed by the transformed question were returned as answers to the original question.
We implemented all three of these answering strategies into a previously existing Q/A system described in (Harabagiu *et al.* 2005). They each correspond to a different text fusion operator.

## 4. Recognizing Contrasts

Several discourse theories recognize CONTRAST as a discourse relation between discourse units which can be either separate sentences or separate clauses within the same sentence. As described in (Kehler 2002), CONTRAST is a resemblance discourse relation that can be recognized by the presence of the cue phrases "*but*", or "*although*". Figure 4(a) illustrates an example of a CONTRAST relation that was introduced in (Marcu & Echihabi 2002). In the example, the presence of *opposing* information contributes more to the assessment of a CONTRAST than the presence of a cue phrase. More specifically, by recognizing the relation of *antonymy* between the expressions "*preclude*" and "*be able*" or between "*embargo*" and "*legally*" in the example illustrated in Figure 4(a), we are able to conclude that there is a CONTRAST relation between the two sentences, without needing to take the cue phrase into account.

### 4.1 Discovering CONTRAST Relations

We argue that each discourse relation can be characterized by specific semantic and pragmatic signatures. Cue phrases,

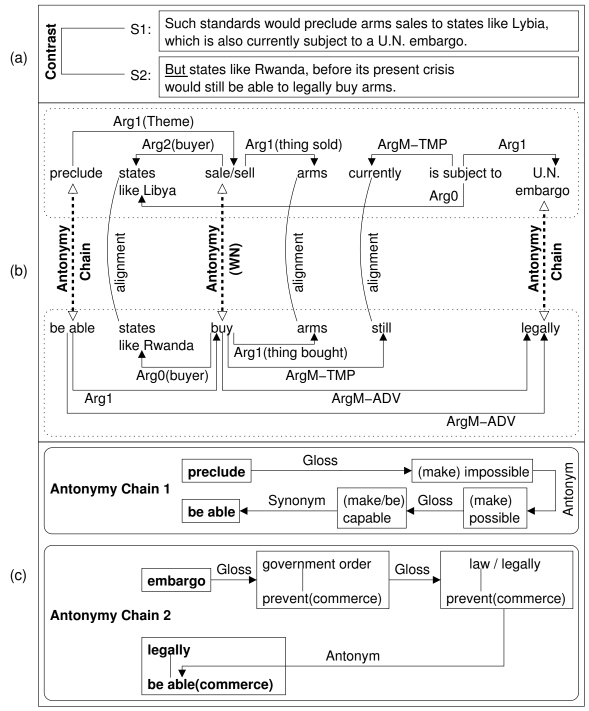


Figure 4: Contrast Relation Indicated by Antonymy Chains.

when present, signal a certain discourse relation. However, even in the absence of cue phrases, humans can recognize discourse relations, in large part due to world knowledge. By assuming that two discourse units that establish a relation must have (a) some common information they share and (b) some information that justifies the relation, we have devised the following procedure for recognizing CONTRAST relations:

◇ STEP 1. Generate predicate-argument structures.
◇ STEP 2. Find possible alignments between predicates and arguments.
◇ STEP 3. Acquire paraphrases of aligned information.
◇ STEP 4. Cluster paraphrases into sets conveying the same concept (using complete-link clustering).
◇ STEP 5. Compute the alignment probability and alignment features.
◇ STEP 6. Compute paraphrase features, dependency features, contrast features and semantic / pragmatic features. The features are illutrated in Figure 5.
◇ STEP 7. Use a Maximum Entropy classifier to decide the recognition of a CONTRAST relation.

As Figure 4(b) illustrates, predicate-argument structures determined by semantic parsers trained on PropBank or NomBank enable the thematic alignment between texts. There are five forms of alignment links between predicate-argument structures:
(1) alignment between arguments having identical heads (e.g. "*states*" (like Libya) and "*states*" (like Rwanda));
(2) antonymy relations encoded in WordNet (e.g. "*sell*" and "*buy*");
(3) antonymy chains that are derived from WordNet for approximating opposing information;
(4) alignment based on functional similarity or inclusion. For example, when the arguments have a specialized relation to the predicate (e.g. temporal) and the function of one of the arguments is similar or includes the function of the other argument. For example, a temporal relation marked by the adverb "*currently*" indicates that the predicate argument function is constrained by a temporal function $F_1$ indicating veridicity at the present time. The temporal function $F_2$ indicated by the adverb "*still*" consists of veridicity not only at the present time, but also in the recent past. Thus, function $F_2$ includes $F_1$.
(5) synonym chains that connect synonymous words.

The recognition of CONTRAST relations based on the ANTONYM relation suffers from the fact that the only available resource that encodes such relations, namely WordNet, considers only pairs of words that share the same part-of-speech (PoS). In WordNet there are 1713 antonyms between nouns, 1025 antonyms between verbs, 3748 antonyms between adjectives, and 704 antonyms between adverbs. As noted in (Marcu & Echihabi 2002) by relying only on antonyms accessible from WordNet we could not account for the antonymy between "*embargo*" and "*legally*" in the example illustrated in Figure 4(a). To solve this problem, Marcu and Echihabi have gathered vast amounts of training data that enable them to learn which pairs of lexical items are likely to co-occur in CONTRAST relation, and thus could be antonyms. In our study we have considered an alternative solution, by automatically generating *antonym chains* based on the information available from WordNet and using the features of these chains for training a classifier. In this way, we make use not only of the lexical information from a vast training corpus, but also of the relational semantics established with the help of WordNet. Figure 4(b) illustrates the existence of two antonym chains between the sentences of the example illustrated in Figure 4(a). The antonym chains are illustrated in Figure 4(c), and the methodology used for generating them is presented in Subsection 4.2. As it can be noted, each antonymy chain contains an antonymy relation along with other relations derived from WordNet.

## 4.2 Generating Antonymy Chains

Antonymy chains are lexico-semantic chains that are based on relations encoded in WordNet in which one relation is the *antonymy* relation. In WordNet, there are encoded more than 7,000 antonymy relations. As reported in (Harabagiu & Moldovan 1998), a lexical chain which combines only IS-A relations with antonymy relations preserves the semantics of opposition dictated by antonymy. Additionally, the relation between a WordNet concept and the genus of its gloss is often encoded as an IS-A relation, and thus we allow combination of GLOSS relations and antonyms and antonymy chains. Finally, antonymy is established between predications, and thus we allow a minimal context of a dependency relation to be considered. Dependency relations in concept glosses are processed by the eXtended WordNet (http://xwn.utdallas.edu). Dependency relations in texts can take the form of predicate-argument relations. To build antonymy chains we follow the steps:

◇ STEP 1. Given a pair of words $(w_1, w_2)$ we retrieve all antonyms encoded in WordNet.
◇ STEP 2. By using only GLOSS and IS-A relations find

**ALIGNMENT FEATURES**: These three features are derived from the results of the lexical alignment classification.
◇1◇ Longest Common String: This feature represents the longest contiguous string common to both texts.
◇2◇ Unaligned Chunk: This feature represents the number of chunks in one text that are not aligned with a chunk from the other text
◇3◇ Lexical Entailment Probability: This feature is defined in (Glickman & Dagan 2005).

**DEPENDENCY FEATURES**: These four features are computed from the PropBank-style annotations assigned by the semantic parser.
◇1◇ Entity-Arg Match: This is a boolean feature which fires when aligned entities were assigned the same argument role label.
◇2◇ Entity-Near-Arg Match: This feature is collapsing the arguments $\text{Arg}_1$ and $\text{Arg}_2$ (as well as $\text{Arg}_M$) into single categories for the purpose of counting matches.
◇3◇ Predicate-Arg Match: This boolean feature is flagged when at least two aligned arguments have the same role.
◇4◇ Predicate-Near-Arg Match: This feature is collapsing the arguments $\text{Arg}_1$ and $\text{Arg}_2$ (as well as $\text{Arg}_M$) into single categories for the purpose of counting matches.

**PARAPHRASE FEATURES**: These three features are derived from the paraphrases acquired for each pair.
◇1◇ Single Pattern Match: This is a boolean feature which fired when a paraphrase matched either of the texts.
◇2◇ Both Pattern Match: This is a boolean feature which fired when paraphrases matched both texts.
◇3◇ Category Match: This is a boolean feature which fired when paraphrases could be found from the same paraphrase cluster that matched both texts.

**NEGATION FEATURES**: These three features take advantage of the truth values that the Preprocessor assigns to predicates.
◇1◇ Truth-Value Mismatch: This is a boolean feature which fired when two aligned predicates differed in any truth value.
◇2◇ Polarity Mismatch: This is a boolean feature which fired when predicates were assigned opposite polarity values.
◇3◇ Overt/Implicit Marker: This is a boolean vector that lists whether a word is (part of) an overt negation marker, an implicit negation marker or not.

**SEMANTIC/PRAGMATIC FEATURES**: These six features are extracted by the preprocessing module.
◇1◇ Named Entity Class: This feature has a different value for each of the 150 named entity classes.
◇2◇ Temporal Normalization: This boolean feature is flagged when the temporal expressions are normalized to the same ISO 9000 equivalents.
◇3◇ Modality Marker: This boolean feature is flagged when the two texts use the same modal verbs.
◇4◇ Speech-Act: This boolean feature is flagged when the lexicons indicate the same speech act in both texts.
◇5◇ Factivity Marker: This boolean feature is flagged when the factivity markers indicate either TRUE or FALSE in both texts simultaneously.
◇6◇ Belief Marker: This boolean feature is flagged when the belief markers indicate either TRUE or FALSE in both texts simultaneously.

**CONTRAST FEATURES**: These six features are derived from the opposing information provided by antonymy relations or chains.
◇1◇ Number of Lexical Antonymy Relations: This feature counts the number of antonyms from WordNet that are discovered between the two texts.
◇2◇ Number of Antonymy Chains: This feature counts the number of antonymy chains that are discovered between the two texts.
◇3◇ Chain Length: This feature represents a vector with the lengths of the antonymy chains that are discovered between the two texts.
◇4◇ Number of Glosses: This feature is a vector representing the number of Gloss relations used in each antonymy chain.
◇5◇ Number of Morphological Changes: This feature is a vector representing the number of Morphological-Derivation relations found in each antonymy chain.
◇6◇ Number of Nodes with Dependencies: This feature is a vector indexing the number of nodes in each antonymy chain that contain dependency relations.

Figure 5: Features Used for Classifying Contradictions.

lexical chains from each of the words $w_i$ to one of the arguments of the antonymy relation.
◇ Step 3. Limit the length of the chain to the antonymy arguments to three.
◇ Step 4. Within a gloss, if a dependency relation is considered, a dependency node is created and the dependency is propagated throughout the chain.
◇ Step 5. When both words are connected to the arguments of the antonyms, a chain is established.

## 5. Evaluation

In this section, we describe how we evaluated the systems for antonymy, contrast, and contradiction detection described in this paper. In addition, we present results from experiments with an automatic question-answering system which shows that input from these three modules can be used to obtain accurate answers to four different types of questions involving negation.

We evaluated our antonymy detection system on a set of 1410 pairs of antonymy extracted from an online thesaurus (wiktionary.org). Each pair of antonyms was submitted to our system separately; if an antonymy chain (of length $l \leq n$) could be found in *WordNet* (WN), the antonym relationship was classified as having been identified correctly. Results were calculated for three types of antonym pairs: (1) pairs where neither term had antonyms listed in WN, (2) pairs where only one term had antonym entries in WN, and (3) pairs where both terms had antonyms in WN. Results from three experiments (with $n$ = 5, $n$ = 8, and $n$ = 10) are presented in Table 2.

| | Examples | $n$=5 | $n$=8 | $n$=10 |
|---|---|---|---|---|
| 0 antonyms in WN | 670 | 0.1791 | 0.3134 | 0.3284 |
| 1 antonym in WN | 460 | 0.2174 | 0.4706 | 0.5000 |
| 2 antonyms in WN | 280 | 0.5185 | 0.5833 | 0.6522 |
| TOTAL | 1410 | 0.2589 | 0.4184 | 0.4489 |

Table 2: Antonym Detection.

These results show that our antonymy detection system was capable of detecting antonymy relations with some accuracy: when comparing pairs of tokens that had no antonym information stored in WN, our algorithm correctly identified antonyms nearly 33% of the time; this number jumped to 65% when both tokens were associated with at least one antonym in WN.

We evaluated our contrast detection system using a corpus of 10,000 instances of the discourse relation CONTRAST extracted from the World Wide Web and newswire documents. Following (Marcu & Echihabi 2002), we considered pairs of adjacent sentences featuring the cue phrase *but* to be positive examples of CONTRAST. (Two types of contexts were considered: (1) pairs of sentences conjoined with *but* (9128 examples), and (2) adjacent sentences where the second sentence included a sentence initial *But* (882 examples).) A total of 9000 sentence pairs were used to train the system; the remaining 1000 pairs were first validated by hand by human annotators and held out as the evaluation set. In order to gather negative examples of CONTRAST, we selected nonadjacent pairs of sentences from each of the documents that positive examples were extracted from. Equal numbers of negative examples were also added to both the training and evaluation sets. Results from our contrast detection system are provided in Table 3.

| | Test Size | Accuracy | Avg. Precision |
|---|---|---|---|
| Contrast | 1200 | 74.19% | 74.00% |

Table 3: Contrast Evaluation.

We believe our performance represents an improvement over (Marcu & Echihabi 2002), as our system classifies pairs of sentences as either positively or negatively representing an instance of CONTRAST; previous approaches were only able to make judgment between two competing discourse relations.

We utilized data prepared for the 2006 PASCAL Recognizing Textual Entailment (RTE) Challenge in order to create three new types of training and evaluation corpora for our textual contradiction system. The 2006 PASCAL RTE dataset consists of 1600 pairs of sentences that have been adapted by human annotators in order to provide 800 examples each of positive and negative textual entailment. [2]

First, in order to test our system's ability to detect examples of contradiction featuring overt negation, we had two annotators negate one sentence from each of the 800 pairs of examples of positive entailment in the RTE dataset in order to create contradictions. (In order to avoid overtraining, annotators were also tasked with adding negative markers to the remaining examples of negative entailment as well; care was taken not to ensure that the resulting example was neither an entailment nor a contradiction.) Examples of a contradiction formed by adding negation is presented in Figure 6.

| |
|---|
| Former dissident John Bok, who has been on a hunger strike since Monday, says he wants to increase pressure on Stanislav Gross to resign as prime minister. |
| A hunger strike was **not** attempted. |

Figure 6: Contradiction by Negation.

Second, in order to evaluate how well our system detected instances of contradiction that did not feature overt negation, human annotators were tasked with creating paraphrases of each of the 800 contradictory sentences created in the first corpus that did not a negative marker. (This was possible in a total of 638 of the 800 examples.) Examples of contradictions formed by paraphrasing a negated sentence is presented in Figure 7

| |
|---|
| Former dissident John Bok, who has been on a hunger strike since Monday, says he wants to increase pressure on Stanislav Gross to resign as prime minister. |
| A hunger strike was **called off**. |

Figure 7: Contradiction by Paraphrasing.

Finally, a third "hybrid" corpus was created by combining 400 examples of contradiction taken randomly from both "negated" corpus and the "paraphrased" corpus and a total of 800 negative examples of contradiction.

We have previously described the evaluation of the textual entailment (TE) system that forms the core of our textual contradiction (TC) system in (Hickl *et al.* 2006). In addition to being trained on the 800 examples found in the 2006 PASCAL RTE Development Set, this system was also trained on over 200,000 examples of textual entailment that were automatically extracted from the newswire corpora and the WWW. While we did not create such a large training corpus for our textual contradiction system, we did use training data acquired for our TE system's *lexical alignment* module in order to train the corresponding alignment module featured in our TC system. Used primarily to determine the likelihood that two constituents encode corresponding lexico-semantic information, the alignment classifier was also used to determine values for the alignment features used in the TC system's final contradiction classifier.

Results for our TC system's performance on each of our three evaluation corpora are presented in Table 4. As in the PASCAL RTE evaluations, two performance metrics are provided: (1) accuracy (equal to the percentage of correct classifications made) and (2) average precision (equal to the system's average confidence in each classification decision) [3].

| | Test Size | Accuracy | Avg. Precision |
|---|---|---|---|
| **Textual Entailment** | | | |
| PASCAL RTE 2006 | 800 | 75.38% | 80.16% |
| **Contradiction** | | | |
| Negation Only | 800 | 75.63% | 68.07% |
| Paraphrase Only | 800 | 62.55% | 67.35% |
| Paraphrase + Negation | 800 | 64.00% | 75.74% |

Table 4: Evaluation of Textual Inference.

When evaluated on the "negated" contradiction corpus, our textual contradiction system correctly classified 75.63% of examples, a result nearly equivalent to our textual entailment system's accuracy (75.38%) on 2006 PASCAL RTE data. This suggests that the method for recognizing textual contradiction presented in View 1 in Figure 3 may prove effective for classifying pairs of texts that differ in polarity, as reasonably high levels of accuracy can be obtained, regardless if the a system processes texts by annotating polarity values or removing negative markers altogether. Despite this promising result, our system's performance dropped to 62.55% on the corpus of "paraphrased" contradictions and to 64.00% on the "hybrid" corpus combining negated and paraphrased contradictions. Although these scores are substantially lower than our previous result, they are both in line with the performance of TE system (65.25%) on PASCAL data when no additional sources of training data were used. We believe that these results demonstrate the viability of the approach outlined in View 2 in Figure 3. In both of these cases, competitive scores were obtained by employing an approach which combined multiple different types of semantic annotations using a robust classifier.

In addition to recognizing contradictions in isolation, we also evaluated the performance of our textual contradiction

[2]Data is publicly available at http://www.pascal-network.org/Challenges/RTE2/Datasets/.

[3]Following (Voorhees 1999), *Average Precision* is calculated as $\frac{1}{R} * \sum_{i=1}^{n} (E(i) * \frac{\#-correct-up-to-pair-i}{i})$ where $n$ is the number of the pairs in the test set, $R$ is the total number of positive pairs in the test set, $E(i)$ is 1 if the $i$-th pair is positive and 0 otherwise, and $i$ ranges over the pairs, ordered by their ranking.

framework in the context of answering negative questions using the automatic question-answering system described in (Harabagiu *et al.* 2005). Table 5 details the performance of four different strategies: (1) a *negation detection* strategy which returned answers associated with a negated predicate, attribute, or entity, (2) a *set intersection* strategy which returned answers that were not included in the intersection of the answers returned by two positive analogs of the question, (3) an *entailment* strategy which returned answers that were deemed to be close approximations of the negated proposition expressed by the question, and (4) a *hybrid* strategy, which combined and re-ranked answers returned by all three strategies using a method described in (Harabagiu *et al.* 2005). In a evaluation of 150 negative questions, the hybrid strategy performed best, returning an answer in top-ranked position to 83 questions (55.33%). The results were evaluated using the Mean Reciprocal Rank (MRR) score (Voorhees 1999) and the Accuracy of the answer. The reciprocal rank (RR) of an answer is defined as (1/Rank$_i$), where $i$ is the position of the first correct answer to the question. The Mean Reciprocal Rank (MRR) is then defined as the mean of the sum of the reciprocal ranks for a set of questions divided by the total number of questions (i.e. $MRR = \sum_{i=1}^{n}(RR_i)/N$).

| **Strategy** | **Accuracy** | **MRR** |
|---|---|---|
| Negation Detection | 24.7% | 0.298 |
| Set Intersection | 17.3% | 0.213 |
| Entailment | 48.0% | 0.404 |
| Hybrid | 55.3% | 0.495 |

Table 5: Answering Negative Questions

## 6. Conclusions

In this paper, we have introduced a new framework for recognizing contradiction in natural language texts. Our approach combines techniques for the processing of negation, the recognition of contrasts, and the automatic detection of antonymy in order to identify instances of contradiction with over 62% accuracy. In addition, we have incorporated each of these components in order to create new types of fusion operators for automatic question-answering systems and new techniques for answering negative questions.

## 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

Bierner, G., and Webber, B. 2000. Inference through Alternative-Set Semantics. *Journal of Language and Computation* 1(2):259–274.

Chapman, W. W.; Bridewell, W.; Hanbury, P.; Cooper, G. F.; and Buchanan, B. G. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics* 34(5):301–310.

Collins, M. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. Dissertation, Univ. of Pennsylvania.

Fellbaum, C., ed. 1998. *Wordnet: An Electronic Lexical Database*. The MIT Press.

Glickman, O., and Dagan, I. 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*.

Greenberg, J. H., ed. 1978. *Universals of human language Vol. 4: Syntax.* Stanford, California: Stanford University Press.

Haghighi, A.; Ng, A.; and Manning, C. D. 2005. Robust textual inference via graph matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-05)*.

Harabagiu, S., and Moldovan, D. 1998. Knowledge Processing on Extended WordNet. In Fellbaum, C., ed., *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press. 379–405.

Harabagiu, S.; Moldovan, D.; Clark, C.; Bowden, M.; Williams, J.; and Bensley, J. 2003. Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In *Proceedings of the Twelfth Text REtrieval Conference*.

Harabagiu, S.; Moldovan, D.; Clark, C.; Bowden, M.; Hickl, A.; and Wang, P. 2005. Employing Two Question Answering Systems in TREC 2005. In *Proceedings of the Fourteenth Text REtrieval Conference*.

Hickl, A.; Williams, J.; Bensley, J.; Roberts, K.; Rink, B.; and Shi, Y. 2006. Recognizing Textual Entailment with LCC's Groundhog System. In *Proceedings of the Second PASCAL Challenges Workshop (to appear)*.

Kehler, A. 2002. *Coherence, Reference, and the Theory of Grammar*. Stanford, CA: CSLI.

Marcu, D., and Echihabi, A. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 4Oth Meeting of the Association for Computational Linguistics*.

Meyers, A.; Reeves, R.; Macleod, C.; Szekely, R.; Zielinska, V.; Young, B.; and Grishman, R. 2004. The nombank project: An interim report. In Meyers, A., ed., *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, 24–31. Boston, Massachusetts, USA: Association for Computational Linguistics.

Palmer, M.; Gildea, D.; and Kingsbury, P. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics* 31(1):71–106.

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

Surdeanu, M.; Harabagiu, S. M.; Williams, J.; and Aarseth, P. 2003. Using predicate-argument structures for information extraction. In *ACL*, 8–15.

Voorhees, E. M. 1999. The TREC-8 question answering track report. In *Proceedings of the 8th Text REtrieval Conference*, 77–82.