Finding Contradictions in Text

Marie-Catherine de Marneffe, **Anna N. Rafferty** and **Christopher D. Manning**
Linguistics Department
Stanford University
Stanford, CA 94305
mcdm@stanford.edu

Computer Science Department
Stanford University
Stanford, CA 94305
{rafferty,manning}@stanford.edu

Abstract

Detecting conflicting statements is a foundational text understanding task with applications in information analysis. We propose an appropriate definition of contradiction for NLP tasks and develop available corpora, from which we construct a typology of contradictions. We demonstrate that a system for contradiction needs to make more fine-grained distinctions than the common systems for entailment. In particular, we argue for the centrality of event coreference and therefore incorporate such a component based on topicality. We present the first detailed breakdown of performance on this task. Detecting some types of contradiction requires deeper inferential paths than our system is capable of, but we achieve good performance on types arising from negation and antonymy.

1 Introduction

In this paper, we seek to understand the ways contradictions occur across texts and describe a system for automatically detecting such constructions. As a foundational task in text understanding (Condoravdi et al., 2003), contradiction detection has many possible applications. Consider applying a contradiction detection system to political candidate debates: by drawing attention to topics in which candidates have conflicting positions, the system could enable voters to make more informed choices between candidates and sift through the amount of available information. Contradiction detection could also be applied to intelligence reports, demonstrating which information may need further verification. In bioinformatics where protein-protein interaction is widely studied, automatically finding conflicting facts about such interactions would be beneficial.

Here, we shed light on the complex picture of contradiction in text. We provide a definition of contradiction suitable for NLP tasks, as well as a collection of contradiction corpora. Analyzing these data, we find contradiction is a rare phenomenon that may be created in different ways; we propose a typology of contradiction classes and tabulate their frequencies. Contradictions arise from relatively obvious features such as antonymy, negation, or numeric mismatches. They also arise from complex differences in the structure of assertions, discrepancies based on world-knowledge, and lexical contrasts.

(1) Police specializing in explosives defused the rockets. Some 100 people were working inside the plant.

(2) 100 people were injured.

This pair is contradictory: defused rockets cannot go off, and thus cannot injure anyone. Detecting contradictions appears to be a harder task than detecting entailments. Here, it is relatively easy to identify the lack of entailment: the first sentence involves no injuries, so the second is unlikely to be entailed. Most entailment systems function as weak proof theory (Hickl et al., 2006; MacCartney et al., 2006; Zanzotto et al., 2007), but contradictions require deeper inferences and model building. While mismatching information between sentences is often a good cue of non-entailment (Vanderwende et al., 2006), it is not sufficient for contradiction detection which requires more precise comprehension of the consequences of sentences. Assessing event coreference is also essential: for texts to contradict, they must

Proceedings of ACL-08: HLT, pages 1039–1047,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

refer to the same event. The importance of event coreference was recognized in the MUC information extraction tasks in which it was key to identify scenarios related to the same event (Humphreys et al., 1997). Recent work in text understanding has not focused on this issue, but it must be tackled in a successful contradiction system. Our system includes event coreference, and we present the first detailed examination of contradiction detection performance, on the basis of our typology.

2 Related work

Little work has been done on contradiction detection. The PASCAL Recognizing Textual Entailment (RTE) Challenges (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007) focused on textual inference in any domain. Condoravdi et al. (2003) first recognized the importance of handling entailment and contradiction for text understanding, but they rely on a strict logical definition of these phenomena and do not report empirical results. To our knowledge, Harabagiu et al. (2006) provide the first empirical results for contradiction detection, but they focus on specific kinds of contradiction: those featuring negation and those formed by paraphrases. They constructed two corpora for evaluating their system. One was created by overtly negating each entailment in the RTE2 data, producing a balanced dataset (LCC_negation). To avoid overtraining, negative markers were also added to each non-entailment, ensuring that they did not create contradictions. The other was produced by paraphrasing the hypothesis sentences from LCC_negation, removing the negation (LCC_paraphrase): *A hunger strike was not attempted → A hunger strike was called off.* They achieved very good performance: accuracies of 75.63% on LCC_negation and 62.55% on LCC_paraphrase. Yet, contradictions are not limited to these constructions; to be practically useful, any system must provide broader coverage.

3 Contradictions

3.1 What is a contradiction?

One standard is to adopt a strict logical definition of contradiction: sentences A and B are contradictory if there is no possible world in which A and B are both true. However, for contradiction detection to be useful, a looser definition that more closely matches human intuitions is necessary; contradiction occurs when two sentences are extremely unlikely to be true simultaneously. Pairs such as *Sally sold a boat to John* and *John sold a boat to Sally* are tagged as contradictory even though it could be that each sold a boat to the other. This definition captures intuitions of incompatiblity, and perfectly fits applications that seek to highlight discrepancies in descriptions of the same event. Examples of contradiction are given in table 1. For texts to be contradictory, they must involve the same event. Two phenomena must be considered in this determination: implied coreference and embedded texts. Given limited context, whether two entities are coreferent may be probable rather than certain. To match human intuitions, compatible noun phrases between sentences are assumed to be coreferent in the absence of clear countervailing evidence. In the following example, it is not necessary that the *woman* in the first and second sentences is the same, but one would likely assume it is if the two sentences appeared together:

(1) Passions surrounding Germany's final match turned violent when a woman stabbed her partner because she didn't want to watch the game.

(2) A woman passionately wanted to watch the game.

We also mark as contradictions pairs reporting contradictory statements. The following sentences refer to the same event (*de Menezes in a subway station*), and display incompatible views of this event:

(1) Eyewitnesses said de Menezes had jumped over the turnstile at Stockwell subway station.

(2) The documents leaked to ITV News suggest that Menezes walked casually into the subway station.

This example contains an "embedded contradiction." Contrary to Zaenen et al. (2005), we argue that recognizing embedded contradictions is important for the application of a contradiction detection system: if *John thinks that he is incompetent*, and *his boss believes that John is not being given a chance*, one would like to detect that the targeted information in the two sentences is contradictory, even though the two sentences can be true simultaneously.

3.2 Typology of contradictions

Contradictions may arise from a number of different constructions, some overt and others that are com-

ID	Type	Text	Hypothesis
1	Antonym	Capital punishment is a catalyst for more crime.	Capital punishment is a deterrent to crime.
2	Negation	A closely divided Supreme Court said that juries and not judges must impose a death sentence.	The Supreme Court decided that only judges can impose the death sentence.
3	Numeric	The tragedy of the explosion in Qana that killed more than 50 civilians has presented Israel with a dilemma.	An investigation into the strike in Qana found 28 confirmed dead thus far.
4	Factive	Prime Minister John Howard says he will not be swayed by a warning that Australia faces more terrorism attacks unless it withdraws its troops from Iraq.	Australia withdraws from Iraq.
5	Factive	The bombers had not managed to enter the embassy.	The bombers entered the embassy.
6	Structure	Jacques Santer succeeded Jacques Delors as president of the European Commission in 1995.	Delors succeeded Santer in the presidency of the European Commission.
7	Structure	The Channel Tunnel stretches from England to France. It is the second-longest rail tunnel in the world, the longest being a tunnel in Japan.	The Channel Tunnel connects France and Japan.
8	Lexical	The Canadian parliament's Ethics Commission said former immigration minister, Judy Sgro, did nothing wrong and her staff had put her into a conflict of interest.	The Canadian parliament's Ethics Commission accuses Judy Sgro.
9	Lexical	In the election, Bush called for U.S. troops to be withdrawn from the peacekeeping mission in the Balkans.	He cites such missions as an example of how America must "stay the course."
10	WK	Microsoft Israel, one of the first Microsoft branches outside the USA, was founded in 1989.	Microsoft was established in 1989.

Table 1: Examples of contradiction types.

plex even for humans to detect. Analyzing contradiction corpora (see section 3.3), we find two primary categories of contradiction: (1) those occurring via antonymy, negation, and date/number mismatch, which are relatively simple to detect, and (2) contradictions arising from the use of factive or modal words, structural and subtle lexical contrasts, as well as world knowledge (WK).

We consider contradictions in category (1) 'easy' because they can often be automatically detected without full sentence comprehension. For example, if words in the two passages are antonyms and the sentences are reasonably similar, especially in polarity, a contradiction occurs. Additionally, little external information is needed to gain broad coverage of antonymy, negation, and numeric mismatch contradictions; each involves only a closed set of words or data that can be obtained using existing resources and techniques (e.g., WordNet (Fellbaum, 1998), VerbOcean (Chklovski and Pantel, 2004)).

However, contradictions in category (2) are more difficult to detect automatically because they require precise models of sentence meaning. For instance, to find the contradiction in example 8 (table 1), it is necessary to learn that *X said Y did nothing wrong* and *X accuses Y* are incompatible. Presently, there exist methods for learning oppositional terms (Marcu and Echihabi, 2002) and paraphrase learning has been thoroughly studied, but successfully extending these techniques to learn incompatible phrases poses difficulties because of the data distribution. Example 9 provides an even more difficult instance of contradiction created by a lexical discrepancy. Structural issues also create contradictions (examples 6 and 7). Lexical complexities and variations in the function of arguments across verbs can make recognizing these contradictions complicated. Even when similar verbs are used and argument differences exist, structural differences may indicate non-entailment or contradiction, and distinguishing the two automatically is problematic. Consider contradiction 7 in table 1 and the following non-contradiction:

(1) The CFAP purchases food stamps from the government and distributes them to eligible recipients.

(2) A government purchases food.

Data	# contradictions	# total pairs
RTE1_dev1	48	287
RTE1_dev2	55	280
RTE1_test	149	800
RTE2_dev	111	800
RTE3_dev	80	800
RTE3_test	72	800

Table 2: Number of contradictions in the RTE datasets.

In both cases, the first sentence discusses one entity (*CFAP*, *The Channel Tunnel*) with a relationship (*purchase*, *stretch*) to other entities. The second sentence posits a similar relationship that includes one of the entities involved in the original relationship as well as an entity that was not involved. However, different outcomes result because a tunnel connects only two unique locations whereas more than one entity may purchase food. These frequent interactions between world-knowledge and structure make it hard to ensure that any particular instance of structural mismatch is a contradiction.

3.3 Contradiction corpora

Following the guidelines above, we annotated the RTE datasets for contradiction. These datasets contain pairs consisting of a short text and a one-sentence hypothesis. Table 2 gives the number of contradictions in each dataset. The RTE datasets are balanced between entailments and non-entailments, and even in these datasets targeting inference, there are few contradictions. Using our guidelines, RTE3_test was annotated by NIST as part of the RTE3 Pilot task in which systems made a 3-way decision as to whether pairs of sentences were entailed, contradictory, or neither (Voorhees, 2008).[1]

Our annotations and those of NIST were performed on the original RTE datasets, contrary to Harabagiu et al. (2006). Because their corpora are constructed using negation and paraphrase, they are unlikely to cover all types of contradictions in section 3.2. We might hypothesize that rewriting explicit negations commonly occurs via the substitution of antonyms. Imagine, e.g.:

H: Bill has finished his math.

	Type	RTE sets	'Real' corpus
1	Antonym	15.0	9.2
	Negation	8.8	17.6
	Numeric	8.8	29.0
2	Factive/Modal	5.0	6.9
	Structure	16.3	3.1
	Lexical	18.8	21.4
	WK	27.5	13.0

Table 3: Percentages of contradiction types in the RTE3_dev dataset and the real contradiction corpus.

Neg-H: Bill hasn't finished his math.

Para-Neg-H: Bill is still working on his math.

The rewriting in both the negated and the paraphrased corpora is likely to leave one in the space of 'easy' contradictions and addresses fewer than 30% of contradictions (table 3). We contacted the LCC authors to obtain their datasets, but they were unable to make them available to us. Thus, we simulated the LCC_negation corpus, adding negative markers to the RTE2 test data (Neg_test), and to a development set (Neg_dev) constructed by randomly sampling 50 pairs of entailments and 50 pairs of non-entailments from the RTE2 development set.

Since the RTE datasets were constructed for textual inference, these corpora do not reflect 'real-life' contradictions. We therefore collected contradictions 'in the wild.' The resulting corpus contains 131 contradictory pairs: 19 from newswire, mainly looking at related articles in Google News, 51 from Wikipedia, 10 from the Lexis Nexis database, and 51 from the data prepared by LDC for the distillation task of the DARPA GALE program. Despite the randomness of the collection, we argue that this corpus best reflects naturally occurring contradictions.[2]

Table 3 gives the distribution of contradiction types for RTE3_dev and the real contradiction corpus. Globally, we see that contradictions in category (2) occur frequently and dominate the RTE development set. In the real contradiction corpus, there is a much higher rate of the negation, numeric and lexical contradictions. This supports the intuition that in the real world, contradictions primarily occur for two reasons: information is updated as knowledge

[1] Information about this task as well as data can be found at http://nlp.stanford.edu/RTE3-pilot/.

[2] Our corpora—the simulation of the LLC_negation corpus, the RTE datasets and the real contradictions—are available at http://nlp.stanford.edu/projects/contradiction.

of an event is acquired over time (e.g., a rising death toll) or various parties have divergent views of an event (e.g., example 9 in table 1).

4 System overview

Our system is based on the stage architecture of the Stanford RTE system (MacCartney et al., 2006), but adds a stage for event coreference decision.

4.1 Linguistic analysis

The first stage computes linguistic representations containing information about the semantic content of the passages. The text and hypothesis are converted to typed dependency graphs produced by the Stanford parser (Klein and Manning, 2003; de Marneffe et al., 2006). To improve the dependency graph as a pseudo-semantic representation, collocations in WordNet and named entities are collapsed, causing entities and multiword relations to become single nodes.

4.2 Alignment between graphs

The second stage provides an alignment between text and hypothesis graphs, consisting of a mapping from each node in the hypothesis to a unique node in the text or to null. The scoring measure uses node similarity (irrespective of polarity) and structural information based on the dependency graphs. Similarity measures and structural information are combined via weights learned using the passive-aggressive online learning algorithm MIRA (Crammer and Singer, 2001). Alignment weights were learned using manually annotated RTE development sets (see Chambers et al., 2007).

4.3 Filtering non-coreferent events

Contradiction features are extracted based on mismatches between the text and hypothesis. Therefore, we must first remove pairs of sentences which do not describe the same event, and thus cannot be contradictory to one another. In the following example, it is necessary to recognize that *Pluto's moon* is not the same as *the moon Titan*; otherwise conflicting diameters result in labeling the pair a contradiction.

T: Pluto's moon, which is only about 25 miles in diameter, was photographed 13 years ago.

H: The moon Titan has a diameter of 5100 kms.

This issue does not arise for textual entailment: elements in the hypothesis not supported by the text lead to non-entailment, regardless of whether the same event is described. For contradiction, however, it is critical to filter unrelated sentences to avoid finding false evidence of contradiction when there is contrasting information about different events.

Given the structure of RTE data, in which the hypotheses are shorter and simpler than the texts, one straightforward strategy for detecting coreferent events is to check whether the root of the hypothesis graph is aligned in the text graph. However, some RTE hypotheses are testing systems' abilities to detect relations between entities (e.g., *John of IBM* ... → *John works for IBM*). Thus, we do not filter verb roots that are indicative of such relations. As shown in table 4, this strategy improves results on RTE data. For real world data, however, the assumption of directionality made in this strategy is unfounded, and we cannot assume that one sentence will be short and the other more complex. Assuming two sentences of comparable complexity, we hypothesize that modeling topicality could be used to assess whether the sentences describe the same event.

There is a continuum of topicality from the start to the end of a sentence (Firbas, 1971). We thus originally defined the topicality of an NP by n^w where n is the nth NP in the sentence. Additionally, we accounted for multiple clauses by weighting each clause equally; in example 4 in table 1, *Australia* receives the same weight as *Prime Minister* because each begins a clause. However, this weighting was not supported empirically, and we thus use a simpler, unweighted model. The topicality score of a sentence is calculated as a normalized score across all aligned NPs.[3] The text and hypothesis are topically related if either sentence score is above a tuned threshold. Modeling topicality provides an additional improvement in precision (table 4).

While filtering provides improvements in performance, some examples of non-coreferent events are still not filtered, such as:

T: Also Friday, five Iraqi soldiers were killed and nine

[3] Since dates can often be viewed as scene setting rather than what the sentence is about, we ignore these in the model. However, ignoring or including dates in the model creates no significant differences in performance on RTE data.

Strategy	Precision	Recall
No filter	55.10	32.93
Root	61.36	32.93
Root + topic	61.90	31.71

Table 4: Precision and recall for contradiction detection on RTE3_dev using different filtering strategies.

wounded in a bombing, targeting their convoy near Beiji, 150 miles north of Baghdad.

H: Three Iraqi soldiers also died Saturday when their convoy was attacked by gunmen near Adhaim.

It seems that the real world frequency of events needs to be taken into account. In this case, attacks in Iraq are unfortunately frequent enough to assert that it is unlikely that the two sentences present mismatching information (i.e., different location) about the same event. But compare the following example:

T: President Kennedy was assassinated in Texas.

H: Kennedy's murder occurred in Washington.

The two sentences refer to one unique event, and the location mismatch renders them contradictory.

4.4 Extraction of contradiction features

In the final stage, we extract contradiction features on which we apply logistic regression to classify the pair as contradictory or not. The feature weights are hand-set, guided by linguistic intuition.

5 Features for contradiction detection

In this section, we define each of the feature sets used to capture salient patterns of contradiction.

Polarity features. Polarity difference between the text and hypothesis is often a good indicator of contradiction, provided there is a good alignment (see example 2 in table 1). The polarity features capture the presence (or absence) of linguistic markers of negative polarity contexts. These markers are scoped such that words are considered negated if they have a negation dependency in the graph or are an explicit linguistic marker of negation (e.g., simple negation (*not*), downward-monotone quantifiers (*no*, *few*), or restricting prepositions). If one word is negated and the other is not, we may have a polarity difference. This difference is confirmed by checking that the words are not antonyms and that they lack unaligned prepositions or other context that suggests they do not refer to the same thing. In some cases, negations are propagated onto the governor, which allows one to see that *no bullet penetrated* and *a bullet did not penetrate* have the same polarity.

Number, date and time features. Numeric mismatches can indicate contradiction (example 3 in table 1). The numeric features recognize (mis-)matches between numbers, dates, and times. We normalize date and time expressions, and represent numbers as ranges. This includes expression matching (e.g., *over 100* and *200* is not a mismatch). Aligned numbers are marked as mismatches when they are incompatible and surrounding words match well, indicating the numbers refer to the same entity.

Antonymy features. Aligned antonyms are a very good cue for contradiction. Our list of antonyms and contrasting words comes from WordNet, from which we extract words with direct antonymy links and expand the list by adding words from the same synset as the antonyms. We also use oppositional verbs from VerbOcean. We check whether an aligned pair of words appears in the list, as well as checking for common antonym prefixes (e.g., *anti*, *un*). The polarity of the context is used to determine if the antonyms create a contradiction.

Structural features. These features aim to determine whether the syntactic structures of the text and hypothesis create contradictory statements. For example, we compare the subjects and objects for each aligned verb. If the subject in the text overlaps with the object in the hypothesis, we find evidence for a contradiction. Consider example 6 in table 1. In the text, the subject of *succeed* is *Jacques Santer* while in the hypothesis, *Santer* is the object of *succeed*, suggesting that the two sentences are incompatible.

Factivity features. The context in which a verb phrase is embedded may give rise to contradiction, as in example 5 (table 1). Negation influences some factivity patterns: *Bill forgot to take his wallet* contradicts *Bill took his wallet* while *Bill did not forget to take his wallet* does not contradict *Bill took his wallet*. For each text/hypothesis pair, we check the (grand)parent of the text word aligned to the hypothesis verb, and generate a feature based on its factivity class. Factivity classes are formed by clustering our expansion of the PARC lists of factive, implicative and non-factive verbs (Nairn et al., 2006) according to how they create contradiction.

Modality features. Simple patterns of modal reasoning are captured by mapping the text and hypothesis to one of six modalities (*(not_)possible*, *(not_)actual*, *(not_)necessary*), according to the presence of predefined modality markers such as *can* or *maybe*. A feature is produced if the text/hypothesis modality pair gives rise to a contradiction. For instance, the following pair will be mapped to the contradiction judgment (*possible*, *not_possible*):

T: The trial court may allow the prevailing party reasonable attorney fees as part of costs.

H: The prevailing party may not recover attorney fees.

Relational features. A large proportion of the RTE data is derived from information extraction tasks where the hypothesis captures a relation between elements in the text. Using Semgrex, a pattern matching language for dependency graphs, we find such relations and ensure that the arguments between the text and the hypothesis match. In the following example, we detect that *Fernandez* works for *FEMA*, and that because of the negation, a contradiction arises.

T: Fernandez, of FEMA, was on scene when Martin arrived at a FEMA base camp.

H: Fernandez doesn't work for FEMA.

Relational features provide accurate information but are difficult to extend for broad coverage.

6 Results

Our contradiction detection system was developed on all datasets listed in the first part of table 5. As test sets, we used RTE1_test, the independently annotated RTE3_test, and Neg_test. We focused on attaining high precision. In a real world setting, it is likely that the contradiction rate is extremely low; rather than overwhelming true positives with false positives, rendering the system impractical, we mark contradictions conservatively. We found reasonable inter-annotator agreement between NIST and our post-hoc annotation of RTE3_test (κ = 0.81), showing that, even with limited context, humans tend to agree on contradictions.[4] The results on the test sets show that performance drops on new data, highlighting the difficulty in generalizing from a small corpus of positive contradiction examples, as well as underlining the complexity of building a broad coverage system. This drop in accuracy on the test sets is greater than that of many RTE systems, suggesting that generalizing for contradiction is more difficult than for entailment. Particularly when addressing contradictions that require lexical and world knowledge, we are only able to add coverage in a piecemeal fashion, resulting in improved performance on the development sets but only small gains for the test sets. Thus, as shown in table 6, we achieve 13.3% recall on lexical contradictions in RTE3_dev but are unable to identify any such contradictions in RTE3_test. Additionally, we found that the precision of category (2) features was less than that of category (1) features. Structural features, for example, caused us to tag 36 non-contradictions as contradictions in RTE3_test, over 75% of the precision errors. Despite these issues, we achieve much higher precision and recall than the average submission to the RTE3 Pilot task on detecting contradictions, as shown in the last two lines of table 5.

	Precision	Recall	Accuracy
RTE1_dev1	70.37	40.43	–
RTE1_dev2	72.41	38.18	–
RTE2_dev	64.00	28.83	–
RTE3_dev	61.90	31.71	–
Neg_dev	74.07	78.43	75.49
Neg_test	62.97	62.50	62.74
LCC_negation	–	–	75.63
RTE1_test	42.22	26.21	–
RTE3_test	22.95	19.44	–
Avg. RTE3_test	10.72	11.69	–

Table 5: Precision and recall figures for contradiction detection. Accuracy is given for balanced datasets only. 'LCC_negation' refers to performance of Harabagiu et al. (2006); 'Avg. RTE3_test' refers to mean performance of the 12 submissions to the RTE3 Pilot.

[4] This stands in contrast with the low inter-annotator agreement reported by Sanchez-Graillet and Poesio (2007) for contradictions in protein-protein interactions. The only hypothesis we have to explain this contrast is the difficulty of scientific material.

	Type	RTE3_dev	RTE3_test
1	Antonym	25.0 (3/12)	42.9 (3/7)
	Negation	71.4 (5/7)	60.0 (3/5)
	Numeric	71.4 (5/7)	28.6 (2/7)
2	Factive/Modal	25.0 (1/4)	10.0 (1/10)
	Structure	46.2 (6/13)	21.1 (4/19)
	Lexical	13.3 (2/15)	0.0 (0/12)
	WK	18.2 (4/22)	8.3 (1/12)

Table 6: Recall by contradiction type.

7 Error analysis and discussion

One significant issue in contradiction detection is lack of feature generalization. This problem is especially apparent for items in category (2) requiring lexical and world knowledge, which proved to be the most difficult contradictions to detect on a broad scale. While we are able to find certain specific relationships in the development sets, these features attained only limited coverage. Many contradictions in this category require multiple inferences and remain beyond our capabilities:

T: The Auburn High School Athletic Hall of Fame recently introduced its Class of 2005 which includes 10 members.

H: The Auburn High School Athletic Hall of Fame has ten members.

Of the types of contradictions in category (2), we are best at addressing those formed via structural differences and factive/modal constructions as shown in table 6. For instance, we detect examples 5 and 6 in table 1. However, creating features with sufficient precision is an issue for these types of contradictions. Intuitively, two sentences that have aligned verbs with the same subject and different objects (or vice versa) are contradictory. This indeed indicates a contradiction 55% of the time on our development sets, but this is not high enough precision given the rarity of contradictions.

Another type of contradiction where precision falters is numeric mismatch. We obtain high recall for this type (table 6), as it is relatively simple to determine if two numbers are compatible, but high precision is difficult to achieve due to differences in what numbers may mean. Consider:

T: Nike Inc. said that its profit grew 32 percent, as the company posted broad gains in sales and orders.

H: Nike said orders for footwear totaled $4.9 billion, including a 12 percent increase in U.S. orders.

Our system detects a mismatch between *32 percent* and *12 percent*, ignoring the fact that one refers to *profit* and the other to *orders*. Accounting for context requires extensive text comprehension; it is not enough to simply look at whether the two numbers are headed by similar words (*grew* and *increase*). This emphasizes the fact that mismatching information is not sufficient to indicate contradiction.

As demonstrated by our 63% accuracy on Neg_test, we are reasonably good at detecting negation and correctly ascertaining whether it is a symptom of contradiction. Similarly, we handle single word antonymy with high precision (78.9%). Nevertheless, Harabagiu et al.'s performance demonstrates that further improvement on these types is possible; indeed, they use more sophisticated techniques to extract oppositional terms and detect polarity differences. Thus, detecting category (1) contradictions is feasible with current systems.

While these contradictions are only a third of those in the RTE datasets, detecting such contradictions accurately would solve half of the problems found in the real corpus. This suggests that we may be able to gain sufficient traction on contradiction detection for real world applications. Even so, category (2) contradictions must be targeted to detect many of the most interesting examples and to solve the entire problem of contradiction detection. Some types of these contradictions, such as lexical and world knowledge, are currently beyond our grasp, but we have demonstrated that progress may be made on the structure and factive/modal types.

Despite being rare, contradiction is foundational in text comprehension. Our detailed investigation demonstrates which aspects of it can be resolved and where further research must be directed.

Acknowledgments

This paper is based on work funded in part by the Defense Advanced Research Projects Agency through IBM and by the Disruptive Technology Office (DTO) Phase III Program for Advanced Question Answering for Intelligence (AQUAINT) through Broad Agency Announcement (BAA) N61339-06-R-0034.

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 on Recognising Textual Entailment*, Venice, Italy.

Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and Christopher D. Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP-04*.

Cleo Condoravdi, Dick Crouch, Valeria de Pavia, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. *Workshop on Text Meaning (2003 May 31)*.

Koby Crammer and Yoram Singer. 2001. Ultraconservative online algorithms for multiclass problems. In *Proceedings of COLT-2001*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In Quinonero-Candela et al., editor, *MLCW 2005, LNAI Volume 3944*, pages 177–190. Springer-Verlag.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-06)*.

Christiane Fellbaum. 1998. *WordNet: an electronic lexical database*. MIT Press.

Jan Firbas. 1971. On the concept of communicative dynamism in the theory of functional sentence perspective. *Brno Studies in English*, 7:23–47.

Danilo Giampiccolo, Ido Dagan, Bernardo Magnini, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast, and contradiction in text processing. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*.

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 on Recognising Textual Entailment*.

Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proceedings of the Workshop on Operational Factors in Pratical, Robust Anaphora Resolution for Unrestricted Texts, 35th ACL meeting*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics*.

Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of the North American Association of Computational Linguistics (NAACL-06)*.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of ICoS-5*.

Olivia Sanchez-Graillet and Massimo Poesio. 2007. Discovering contradiction protein-protein interactions in text. In *Proceedings of BioNLP 2007: Biological, translational, and clinical language processing*.

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 on Recognising Textual Entailment*.

Ellen Voorhees. 2008. Contradictions and justifications: Extensions to the textual entailment task. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

Annie Zaenen, Lauri Karttunen, and Richard S. Crouch. 2005. Local textual inference: can it be defined or circumscribed? In *ACL 2005 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. 2007. Shallow semantics in fast textual entailment rule learners. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.