Identifying Conflicting Information in Texts

Marie-Catherine de Marneffe
Linguistics Dept
Stanford University
Stanford, CA 94305
mcdm@stanford.edu

Anna N. Rafferty[*]
Computer Science Dept
UC Berkeley
Berkeley, CA 94720
rafferty@cs.berkeley.edu

Christopher D. Manning
Computer Science Dept
Stanford University
Stanford, CA 94305
manning@stanford.edu

Abstract

Understanding relationships between text passages is key for information analysis. We focus here on the contradiction relationship, and build a system to detect conflicting statements. We show that such a system needs to make more fine-grained distinctions than the common systems for entailment. Also, we argue for the centrality of event coreference and therefore incorporate a component based on topicality. We propose a typology of contradictions that naturally arise in text, and give the first detailed breakdown of performance for the contradiction detection task. Although detecting some types of contradiction requires deeper inferential paths than our system is capable of, we achieve good performance on types arising from negation and antonymy.

1 Introduction

The primary goal of distillation is to identify information relevant to a user's query. To do so, a system needs to gather pertinent and concise information about the topic in question. A crucial step in this process is to understand the relationships between pieces of text which are retrieved: to choose which information seems relevant, it matters to know whether text snippets are unrelated, equivalent, redundant, or whether they present information that conflicts. So far little work has targeted the notion of "contradiction" in NLP tasks (Harabagiu et al., 2006), however as pointed out by Condoravdi et al. (2003), handling contradiction is necessary to achieve text understanding. We argue that it is also central to information analysis. It is particularly important for analysts to be made aware of conflicting factual claims and divergent viewpoints, which might reflect different sources, political leanings, or even disinformation. Consider the following texts:

(1) Maitur Rehman, a 29-year-old Pakistani from Multan in Punjab, is reported to be the present amir of Jundullah. He had previously served in the Lashkar-e-Jhangvi, an anti-Shia terrorist organisation.

(2) Intelligence sources in the U.S. and Pakistan tell NBC News that Maitur Rehman is a low-level militant operating in South Waziristan.

If one wants to know what is the status of Maitur Rehman, it would be appropriate to retrieve both passages, and let the user know that the information found diverges: if Maitur Rehman is an amir (a leader), then he is not a low-level militant. The pieces of text contain conflicting information, and determining the exact status of Maitur Rehman would demand further analysis. *Relevant* information gathering can benefit from recognizing such contradictions in text.

In this paper, we analyze the nature of conflicting information which usually appears in text, and describe a system to automatically identify these "contradictions". Information conflicts appear when facts diverge: a rising death toll for example, if detached from a time line, shows contradictory figures:

(3) The explosion in Qana that killed more than 50 civilians has presented Israel with a dilemma.

(4) An investigation into the strike in Qana found 78 confirmed dead thus far.

[*] Work done while working at the Stanford NLP Group.

Conflicting opinions are also very frequent in texts: one example is the pair of sentences in (1) and (2), another example is the following where two different sources report conflicting data.

(5) That police statement reinforced published reports, that eyewitnesses said de Menezes had jumped over the turnstile at Stockwell subway station and was wearing a padded jacket, despite warm weather.

(6) However, the documents leaked to ITV News suggest that Menezes, an electrician, walked casually into the subway station and was wearing a light denim jacket.

Distillation would benefit greatly from recognizing differences in opinions: it can indeed matter to a user that the embedded information about *de Menezes* in the two texts above differ. We therefore aim at finding contradictions defined in a broad sense: two pieces of text are contradictory if they are extremely unlikely to be considered true simultaneously (de Marneffe et al., 2008). This definition captures the two cases of conflicting information that typically arise in text: divergent facts and different opinions. In this paper, we describe a system which, given pairs of passages called text (T) and hypothesis (H), decides whether or not they are contradictory. A contradiction system is a key component achieving relevant information gathering, and as such, could be integrated in a global system targeting distillation.

2 System description

Our system to detect contradiction is an adaption of the Stanford RTE (Recognizing Textual Entailment) system (MacCartney et al., 2006). It follows the Stanford system multi-stage architecture. The first stage computes the 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). The second stage provides an alignment between the graphs, consisting of a mapping from each node in the hypothesis to a unique node in the text or to null. Details about the scoring alignment measure and the search algorithm can be found in (de Marneffe et al., 2007; Chambers et al., 2007). In the final stage, we extract contradiction features on which we apply logistic regression to classify the pair as contradictory or not. Features weights are hand-set, guided by linguistic intuition.

Contradiction features rely on mismatches between the text and the hypothesis. However pairs of sentences which do not describe the same event, and thus cannot be contradictory to one another, could nonetheless contain mismatching information. An extra stage to filter non-coreferent events is therefore added before feature extraction. For example, in the following pair, it is necessary to recognize that *the Jonhstown Flood* has nothing to do with *a ferry sinking*; otherwise conflicting death tolls result in labeling the pair a contradiction.

T: More than 2,000 people lost their lives in the devastating Johnstown Flood.

H: 100 or more people lost their lives in a ferry sinking.

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.

2.1 Filtering non-coreferent events

Right now the system uses a crude filter based on topicality. Assuming two sentences of comparable complexity, we hypothesize that modeling topicality could be used to assess whether the sentences describe the same event. The topicality score of a sentence is calculated as a normalized score across all aligned NPs. The text and hypothesis are topically related if either sentence score is above a tuned threshold. While filtering provides improvement in performance (6.8% in precision), some examples of non-coreferent events are still not filtered, such as:

T: Also Friday, five Iraqi soldiers were killed and nine 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: Princess Diana died in Paris.

H: The car accident killing Princess Diana occurred in London.

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

2.2 Contradiction features

Mismatching information between sentences is often a good cue of non-entailment (Vanderwende et al., 2006), but it is not sufficient for contradiction detection which requires more precise comprehension of the consequences of sentences. Some of the features used in the Stanford RTE system have been more precisely defined to only capture mismatches in similar contexts, instead of global mismatching. These features are described below.

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.

Polarity features. Polarity difference between the text and hypothesis is often a good indicator of contradiction, provided there is a good alignment:

T: A closely divided Supreme Court said that juries and not judges must impose a death sentence.

H: The Supreme Court decided that only judges can impose the death sentence.

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 (see example (3–4) above). 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.

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:

T: Jacques Santer succeeded Jacques Delors as president of the European Commission in 1995.

H: Delors succeeded Santer in the presidency of the European Commission.

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:

T: The bombers had not managed to enter the embassy.

H: The bombers entered the embassy.

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 nonfactive 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.

3 Typology of contradictions

The corpora we used to develop our system are the RTE datasets (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007) that we annotated for contradiction. We found that contradictions constitute approximately 10% of these corpora. Since the RTE datasets were constructed for textual inference, they might not reflect 'real-life' contradictions. We therefore also collected contradictions 'in the wild.' The resulting corpus contains 131 contradictory pairs, available at http://nlp.stanford.edu/projects/contradiction. About 40% of the pairs were taken from the GALE distillation data. We excluded repeated pairs. We therefore took 8 pairs out of the 45 "contradicting" pairs in the phase-1 data, and 45 out of the 72 pairs judged contradictory in the phase-2 data.

Analyzing the data, we find two primary categories of contradiction: (1) those occurring via antonymy, negation, and date/number mismatch, and (2) contradictions arising from the use of factive or modal words, structural and subtle lexical contrasts, as well as world knowledge (WK). Category (1) contradictions are more often surface contradictions, which are relatively easy to detect, as in the death toll example (3–4) or in the *de Menezes* example (5–6). The contradictions in the second category are more difficult to find automatically: they involve lexical and structural discrepancies, as well as inconsistency via world knowledge. In (1–2) for instance, the meaning of *amir* is crucial for detecting the contradiction. In the following pair, one needs to have some knowledge about head companies and branches:

(5) Microsoft Israel, one of the first branches outside the USA, was founded in 1989.

(6) Microsoft was established in 1989.

4 Results and discussion

Our contradiction detection system was developed on all datasets listed in the first part of table 1. As test sets, we used RTE1_test as well as the RTE3_test independently annotated by NIST (Voorhees, 2008). 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 ($\kappa = 0.81$), showing that, even with limited context, humans tend to agree on contradictions.[1] 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

[1] 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.

	Precision	Recall
RTE1_dev1	70.37	40.43
RTE1_dev2	72.41	38.18
RTE2_dev	64.00	28.83
RTE3_dev	61.90	31.71
RTE1_test	42.22	26.21
RTE3_test	22.95	19.44
Avg. RTE3_test	10.72	11.69

Table 1: Precision and recall figures for contradiction detection. 'Avg. RTE3_test' refers to mean performance of the 12 submissions to the RTE3 Pilot.

	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 2: Recall by contradiction type.

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 2, 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 1. In the RTE3 Pilot task, systems made a 3-way decision as to whether pairs of sentences were entailed, contradictory, or neither (Voorhees, 2008).

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 2. 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 2), 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.

We handle single word antonymy with high precision (78.9%), as well as negation. Nevertheless, Harabagiu et al. (2006)'s performance on detecting contradictions arising from negation and antonyms (64% on a balanced dataset) 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. Since more than half of the examples found in the real corpus are of that category (60% if we restrict the corpus to the pairs coming from the GALE data), it 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 detection is foundational for information analysis. Our detailed investigation demonstrates which aspects of it can currently be resolved and where further research must be directed.

Acknowledgments

We thank Sebastian Padó for all his work as part of the Stanford Distillation team. 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. The content does not necessarily reflect the views of the U.S. Government, and no official endorsement should be inferred.

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*.

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)*.

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*. 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*.

Marie-Catherine de Marneffe, Trond Grenager, Bill MacCartney, Daniel Cer, Daniel Ramage, Chloé Kiddon, and Christopher D. Manning. 2007. Aligning semantic graphs for textual inference and machine reading. In *AAAI Spring Symposium at Stanford*.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

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*.

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*.

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*.