

June 7, 2008

RECOGNIZING TEXTUAL ENTAILMENT USING LOGICAL INFERENCE: A SURVEY OF THE PASCAL RTE CHALLENGE

Yaroslav Riabinin

Department of Computer Science

University of Toronto

Toronto, ON M5S 3G4, Canada

yaroslav@cs.toronto.edu

ABSTRACT

This paper introduces the task of recognizing textual entailment, which is applicable across multiple NLP domains, because it addresses the variability of semantic expression in natural language. The PASCAL RTE Challenge is presented, along with a description of the datasets and the evaluation procedures used. The growth and evolution of the challenge during its three-year existence is also discussed. What follows is an exploration of submissions that used a logic prover to detect entailment, with an emphasis on how these systems overcame the need for a large amount of background knowledge. Then, several hybrid systems that incorporated logical inference into their design to improve performance are considered. The paper concludes with a discussion of possibilities for future research on recognizing textual entailment.

1. INTRODUCING TEXTUAL ENTAILMENT

The difficulty of processing natural language computationally can often be attributed to the fact that the same *meaning* can be inferred from different linguistic expressions. This problem is further complicated by the fact that one linguistic expression can have multiple meaningful interpretations, especially if different contexts are considered.

This dual ambiguity is a fundamental property of natural language. Consequently, any research domain in which semantic inference is necessary must inevitably address the variability of language. Given the current demand for text processing applications, most sub-fields of Computational Linguistics require some form of semantic inference. However, rather than collaborating on a solution, researchers within each application area tend to work independently to develop methods of handling linguistic ambiguity. While this approach may yield sufficient results for the time being, it is highly inefficient in the long term, since the lack of a unified research community makes it likely that progress in one application domain will not transfer to other domains. Hence, what is needed is a domain-independent task that could be used to compare and evaluate proposed semantic inference models, such that success on this task would simultaneously benefit a variety of NLP sub-fields – Question Answering, Information Extraction, and Machine Translation, to name a few.

It is hypothesized by Bar-Haim et al. [2] that the task of recognizing textual entailment would serve as a unifying generic framework for modeling semantic inference. This task is defined as follows: “Textual entailment recognition is the task of deciding, given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text” [2]. More formally, a text T entails another text H (hypothesis) if the meaning of H can be inferred from the meaning of T by any individual, assuming a common understanding of language and common background knowledge among people. For example, let us suppose that T and H are the following:

T : *WalMart defended itself in court today against claims that its female employees were kept out of jobs in management because they are women*

H : *WalMart was sued for sexual discrimination*

The task, then, is to correctly determine that H is indeed entailed by T . While this may be simple for a human evaluator, it is quite difficult to perform computationally.

The rest of this paper is structured as follows: I will introduce the PASCAL Recognizing Textual Entailment Challenge in Section 2 and describe how it evolved over the three years it has been in existence. Then, in Section 3, I will consider submissions that used First-Order Logic theorem proving for the RTE task. In Section 4, I will discuss systems that incorporated logical inference as only part of their method, in hopes of improving their performance by adding a deeper semantic analysis. I will comment on the ultimate goal of the competition in Section 5, as I conclude the paper and hypothesize about future directions for research in this area.

2. THE NATURE OF THE CHALLENGES

The essence of the competition is best communicated by the organizers of the first PASCAL RTE challenge: “The Recognising Textual Entailment (RTE) Challenge is an attempt to promote an abstract generic task that captures major semantic inference needs across applications” [6]. In response to the demand for a unified and domain-independent framework that has been described in the previous section, a group of researchers have devised a challenge that would motivate other researchers to address the problem of semantic inference on a more abstract level.

Based on the task of recognizing textual entailment, a competition has been held every year since 2004/2005, attracting teams of researchers to submit their systems for evaluation. All submissions are tested on the same dataset, so that a comparison of their performances can be conducted. Three such challenges have been held to this date. What follows is a more detailed overview of these challenges.

2.1 The First Challenge

The main goal of the first PASCAL RTE Challenge (RTE-1) was to establish meaningful baselines for the entailment recognition task, since no previous attempts to do this have been made by anyone. Another objective was to determine the capabilities of current systems. Hence, the general setting of the challenge was more explorative rather than competitive.

The data used for evaluation was manually gathered by the organizers from various sources. It consisted of Text-Hypothesis ($T-H$) pairs of text snippets from the news domain, where T typically consisted of one or two sentences and H was usually made to be a shorter sentence. Sample text pairs were collected from seven subsets, corresponding to seven different application domains. These are as follows: Information Retrieval (IR), Comparable Documents (CD), Reading Comprehension (RC), Question Answering (QA), Information Extraction (IE), Machine Translation (MT), and Paraphrase Acquisition (PP). Human annotators then judged each pair as either being a positive example of entailment (*True*) or a negative one (*False*), dividing the entire dataset into a Development set of 567 examples and a Test set of 800 examples, with a 50%-50% split of *True/False* pairs.

The task was to classify each $T-H$ pair as either being an example of entailment or not. Furthermore, systems could provide a confidence score (between 0 and 1) for their classification, which allowed for a ranking of pairs. Therefore, systems could be evaluated on their *Accuracy* (fraction of correct responses), as well as their *Confidence-Weighted Score* (*cws*), which “rewards the systems’ ability to assign a higher confidence score to the correct judgments than to the wrong ones” [6].

Overall, the results of the 17 submissions were hardly better than the natural baseline, which predicts *True* (or *False*) all the time, or the random baseline that predicts *True* or *False* at random. System accuracies ranged from 50% to 60%, while *cws* scores were as high as 70%. There was considerable variation in the performance between different application domains – for example, the Comparable Documents (CD) task yielded the best results, with some systems achieving an accuracy of 87% and *cws* of 95%.

In order to achieve the best possible performance, competitors incorporated various levels of inference into their systems, ranging from simple world overlap detection to syntactic matching and even complex logical inference. However, Dagan et al. [6] noted that “system complexity and sophistication of inference did not correlate fully with performance, where some of the best results were obtained by rather naïve lexically-based systems”. Despite this conclusion, the authors also pointed out that applied NLP research seems to be progressing towards deeper semantic analysis, since 5 of the submitted systems were already using logical provers.

2.2 The Second Challenge

The main purpose of the second PASCAL RTE Challenge (RTE-2) was to foster the promotion of research on textual entailment. However, the organizers of this challenge also focused on creating a new dataset that had more “realistic” text-hypothesis examples, based mostly on outputs of actual systems [2]. Ultimately, it consisted of 1600 *T-H* pairs, divided equally into a Development and a Test set (each containing 800 pairs), with a 50%-50% split of *True/False* examples, just as in RTE-1. However, in this new challenge, only four of the original application domains were considered. They are as follows: Information Retrieval (IR), Information Extraction (IE), Question Answering (QA), and multi-document summarization (SUM), which is equivalent to the CD domain in RTE-1.

The evaluation measures used in the second challenge were nearly identical to those in RTE-1, except that the *Confidence-Weighted Score* was substituted by *Average Precision*, which evaluates the ranked pairs in a slightly different manner.

The results of this challenge were significantly better than those of the previous one. In particular, two submissions stood out as performing 10% higher than the other systems. The highest accuracy score they achieved was 75.4%, while the highest average precision was 80.8%. The authors of these systems also participated in the next (third) challenge and their work will be discussed in greater detail later in this paper. However, as noted by Bar-Haim et al. [2], both top competitors performed a deeper analysis of the *T-H* pairs, either using a large entailment corpus for training or

developing a logical inference system that relies on extensive linguistic and general background knowledge.

2.3 The Third Challenge

Following the success of the previous two challenges, the third PASCAL RTE Challenge (RTE-3) was structured in a manner that was similar to its predecessors, but with several new innovations. First, longer texts were introduced into the dataset, in order to move towards more comprehensive scenarios. These texts were up to one paragraph in length and were labeled 'L'. They made up roughly 17% of the test set. Otherwise, the data and the evaluation measures used were of the same format as in RTE-2.

Another addition to this year's challenge was a *resource pool* where competitors could share and comment on various resources they used, such as general NLP tools (POS-taggers, parsers, etc.) and background knowledge for logical inference.

Finally, two new and optional pilot tasks were introduced into the challenge:

- (1) Differentiating *unknown* entailments from negative ones. More specifically, systems could label a *T-H* pair as "UNKNOWN", rather than having to make a binary YES/NO distinction in every case.
- (2) Providing justifications for entailment decisions. There was no specified format for this task and success was measured subjectively by human evaluators.

Out of the 26 teams that participated in this challenge, there was one clear winner, as well as a second team that was a close runner-up. These were virtually the same two teams that achieved the best results in RTE-2. As reported in [7] by Dagan et al., Hickl & Bensley [9] achieved the best accuracy with 80%, which was 8% higher than Tatu & Moldovan's [12] second-place performance of 72%.

It has been concluded by organizers of this third challenge that a deeper semantic analysis is beneficial for determining textual entailment [7]. The same conclusion was reached by organizers of the second challenge. It is not surprising, then, that both of the aforementioned top RTE-3 competitors incorporated logical inference and large entailment corpora into their systems. In the next section, I will examine both early and recent attempts to use a logic prover for the RTE task. In particular, I will demonstrate that success with this approach is closely linked to the authors' ability to generate sufficient background knowledge for the proof to be completed.

3. TOWARDS LOGICAL INFERENCE

The importance of logical inference had first been noted in the proceedings of the second PASCAL RTE challenge: “The results show, for the first time, that systems that rely on deep analysis such as syntactic matching and logical inference can considerably outperform lexical systems” [2]. From then on, more teams began to incorporate sophisticated forms of inference into their system design. What follows is an exploration of several submissions that adopted a “purely” logical approach, such that they converted Text-Hypothesis pairs into First-Order Logic and used a theorem prover to decide whether T entails H . However, these systems were not all equally successful on the RTE task. Therefore, the following key differences in their design will be noted:

- (1) how they generated background knowledge for the proof
- (2) how they processed T and H before converting them into logical formulae
- (3) how they decided if T entails H for each Text-Hypothesis pair
- (4) which logic prover they used

3.1 The Basics of a Logical Approach

The work of Akhmatova & Molla [1] can be treated as a baseline for systems that rely on logical inference to detect textual entailment. In fact, that was their goal from the beginning – “to test the practicability of a purely logical approach”. Although their performance on RTE-1 was among the very worst – achieving an accuracy of only 52% - it should be noted that the authors themselves did not expect to exceed the 50% barrier. The reason for this is that their system was designed to address a certain class of entailments – those that were labeled “lexico-syntactic”, which means that they could be handled exclusively with the use of an ideal syntactic parser and an ideal lexical database (if such things existed). It has been estimated by a separate group of researchers that up to 49% of all the entailment pairs in the RTE-1 Development set belong to this category. Hence, one could claim that Akhmatova & Molla could not hope to achieve an accuracy of 50% or higher – but they did, since their system excelled at detecting the absence of entailment, which greatly improved their results. However, this also seems to indicate that their method performed far more poorly than the accuracy suggests, as evidenced by a recall of 10%.

These results could be explained by examining the design of their system. It was intended to be simple – for example, the theorem prover they used was Otter, which was publicly available on the Internet at the time. Furthermore, by their own admission, they used limited background knowledge, which was necessary for Otter to generate proofs correctly. In particular, they hand-coded a very small number of general axioms, such as the fact that “one” could be represented as a word (*one*) or as a numeric

expression (1). The bulk of their world knowledge was extracted from a lexical resource, WordNet, and had to do with the relations that exist between concepts. More specifically, Akhmatova & Molla developed a custom-made measure that approximates the degree of entailment between two arbitrary words, based on hypernym and synset information from WordNet. For example, one could conclude that T entails H if all the concepts in the Hypothesis are synonyms or hypernyms (more general) of concepts in the Text.

It would be fair to claim that the failures of the system could be attributed to the lack of background knowledge, since the logic prover was not able to produce the desired results without the necessary resources. However, prior to the use of Otter, sentences in T and H were divided into minimal semantic elements (*atomic propositions*) using a syntactic parser – Link Parser – which produces an array of links between words. Errors could occur during this pre-processing stage, due to syntactic ambiguity and the difficulty of extracting propositions based solely on the types of links that are identified by the parser.

Furthermore, the system was designed such that every proposition in the Hypothesis is compared to all the propositions in the Text. Entailment is said to hold only if for every proposition in H , there is at least one proposition in T that could entail it. In other words, the input to Otter is a pair of atomic propositions, plus the background knowledge. Even if one could detect entailment between two propositions with perfect accuracy – by having unlimited world knowledge, for example – it seems too stringent to require that *every* proposition in H be entailed by a proposition in T to conclude that T entails H . Sometimes not all the information contained in the Hypothesis is sufficiently relevant to demand entailment from the Text. Also, as discussed previously, if propositions are extracted improperly, this would affect all further decisions made by the system. Perhaps this would account for the low recall of Akhmatova & Molla’s submission – true examples of entailment were not being classified as such, due to the overly restrictive judgment criteria of the method.

3.2 Early Work with COGEX

Fowler et al. [8] also participated in RTE-1 and adopted a purely logical approach to detecting textual entailment. However, they managed to achieve better results than Akhmatova & Molla, which is likely due to the sophistication of their system. Although their overall accuracy was 55%, which is not a significant improvement over their rivals, this measure may be deceiving. The authors manually analyzed *true* entailment pairs in the test set (400 in total) to determine how difficult it is to prove entailment in each case. They identified five levels of difficulty: easy, moderate, difficult, intractable, and invalid.

On the “easy” pairs (81 out of 400), they achieved an accuracy of 85%.

On the “moderate” pairs (122 out of 400), they achieved an accuracy of 58%. Moreover, it was noted that since many of these T - H pairs required external world knowledge, this accuracy could be much improved if a *larger knowledge base* was available. Therefore, the ability of the system to recognize true entailments was more impressive than the overall accuracy score revealed.

This performance may be explained by examining the architecture of their submission. In particular, Fowler et al. implemented the COGEX logic prover, which is a modified version of Otter. Proofs are generated by contradiction, such that the Hypothesis is negated and checked against the Text and anything that can be inferred from the Text. If a contradiction is found, the prover concludes that H is derivable from T , or in other words, T entails H .

In fact, this method is much more intricate – the prover requires two items:

- (1) a “set of support” list of clauses, which consists of all predicates from T and the negated form of H .
- (2) a “usable” list of clauses, which consists of background knowledge.

Once these lists have been obtained, the logic prover removes clauses from the “set of support” list, one by one, and searches the “usable” list for any new inferences that could be made, which are then appended to the “set of support”. This continues until the “set of support” list is empty. Then, if a refutation of the negated Hypothesis has been found during this process, the proof is complete. If not, then predicate arguments are relaxed and another attempt at refutation is made. If this fails, then predicates are dropped from the negated Hypothesis until a contradiction occurs. This technique guarantees that *a proof will always be found*. Hence, to decide whether entailment between T and H exists, a measure is calculated by starting with a perfect score and deducting points for axioms that were used to generate new inferences, arguments that have been relaxed, and predicates that were dropped. If this score is above a certain threshold – empirically calculated from the Development set – then true entailment is established.

It could be argued that Fowler et al.’s success is not attributed to their logic prover, but rather to their generation of background knowledge. The pre-processing of T and H before conversion into logic form included part-of-speech tagging, parse tree generation, and detection of semantic relations. This allowed the system to automatically generate NLP axioms in the form of linguistic re-writing rules that helped break down complex structures into smaller components – for example, by separating clauses or phrases joined by a coordinating conjunction. This sort of syntactic analysis is clearly far less error-prone than Akhmatova & Molla’s method of extracting atomic propositions by looking at syntactic “linkages”.

Furthermore, in addition to a small common-sense knowledge base of axioms that was manually coded by the authors, WordNet lexical chains were used to automatically produce an abundance of new axioms that helped the logic prover infer target concepts from starting concepts. A *lexical chain* is a set of relations between two synsets. If such a lexical chain was found between a synset that contained a word in T and another synset that contained a word in H , then an axiom was generated for each semantic relation in the chain. This allowed the system to capture entailment relations between the *meanings* of the Text and the Hypothesis, resulting in increased performance on the RTE task.

3.3 Recent Success with COGEX

The work of Tatu & Moldovan [12] is quite similar to Fowler et al.'s RTE-1 submission, since they also used the COGEX logic prover to detect textual entailment. This resemblance is likely explained by the fact that both research teams were part of the Language Computer Corporation (LCC). However, Tatu & Moldovan managed to achieve an accuracy score of 72%, which put them into second place at RTE-3. They approached the RTE task as finding logical implication between *meanings*. For this purpose, they first transformed T and H into a semantically-rich logic form, which allowed them to generate an enormous amount of background knowledge that was used to search for an entailment proof between the Text and Hypothesis.

Due to the similarities between this method and Fowler et al.'s [8] work, it may be argued that the success of Tatu & Moldovan can be attributed to the following improvements they made to their system:

- eXtended WordNet Knowledge Base (XWN-KB) – this lexical resource was introduced in order to produce more world knowledge axioms. Specifically, semantic relations were extracted from WordNet *glosses* for each synset using a syntactic and a semantic parser. The authors called the XWN-KB an “invaluable resource for recognizing textual entailment”.
- Coreference Resolution – Fowler et al. mentioned in their work that the “difficult” examples of positive entailment may be handled by adding new functionality, such as coreference resolution [8]. It seems the current authors followed this advice and created a tool that resolved pronouns to their actual referent. Tatu & Moldovan noted that this is especially important for longer texts, where tight connections between predicates are necessary.
- Named Entity Check – when generating NLP axioms in the form of syntactic re-write rules, the system linked named entities to their aliases. For example, “Central Intelligence Agency” was linked to “CIA”. Moreover, the authors introduced a new Entity Check module that deducted points

from the overall proof score if H contained at least one named entity that could not be derived from T . This heuristic was extremely accurate, identifying true negative entailments for 154 out of 167 pairs that it fired on (92% accuracy).

Despite the sophistication of their method, the authors pointed out that numerous errors still occurred due to the failure of their system to generate the correct background knowledge for difficult examples. It is an open question as to whether their system simply lacked complexity or whether it is unrealistic to demand a perfect performance from a machine, given the robustness of natural language.

What follows is a description of RTE submissions that combined logical inference with other NLP techniques to create hybrid systems.

4. BEYOND LOGICAL INFERENCE

The PASCAL RTE Challenges have shown that success on the task of detecting textual entailment is rarely the product of one effective method – rather, it is usually the combination of various NLP techniques that produces the best results. Having examined “purely” logical approaches in the previous section, it may be instructive to evaluate the performance of systems that incorporated some form of logical inference into their design to try to improve their results. It seems the outcome of this endeavour largely depends on the *type* of logical inference that is introduced.

4.1 Augmenting Shallow Methods with First-Order Logic

One of the first efforts to combine shallow NLP methods with a deep semantic analysis was made by Bos & Markert [3] in RTE-1. They began by measuring word overlap between the Text and the Hypothesis and training a decision tree to detect entailment with this feature alone. Then, they generated fine-grained semantic representations for each T - H pair, translated them into First-Order Logic (FOL), and used a theorem prover (Vampire) to check whether an entailment holds. This was done by first producing a set of background knowledge (BK) and determining whether the Text was consistent with this BK. If it was, then an attempt was made to prove entailment with the background knowledge ($BK \wedge T \rightarrow H$). If it was not consistent with the BK, then only ($T \rightarrow H$) was given to the theorem prover.

However, this approach did not yield good results. As the authors explained it, this was “mostly due to the lack of appropriate background knowledge without which many true entailments cannot be found” [3]. This explanation appears to be correct, since Bos & Markert’s BK was composed of only a handful of hand-coded generic axioms (for example, geographical knowledge from the CIA fact-book), in addition to

basic world knowledge axioms in the form of lexical relations derived from WordNet hypernyms. Research on FOL-based theorem proving systems that was presented in the previous section seems to suggest that much more background knowledge is required for such methods to be successful.

In order to improve their performance, Bos & Markert added a finite model builder (Paradox) and used it to compute the model size of $(BK \wedge T)$ and $(BK \wedge T \wedge H)$. They believed that the *difference in the size* of these two models was a good indicator of entailment, since a small difference would mean that the Hypothesis did not introduce any new information, which makes entailment very likely.

Since the deep semantic analysis produced several new features that could be used to describe the T - H pairs, Bos & Markert combined these features with word overlap to train a decision tree to recognize entailment. This increased the performance of their system slightly, yielding a 1% improvement over the accuracy achieved by using the shallow feature alone. As was stated earlier, the authors emphasized that the main reason for their shortcomings was a *lack of background knowledge*. They pointed out that their logical inference method was reasonably accurate on the examples it was able to prove, but it had a small coverage and thus was not very useful.

The second PASCAL RTE Challenge showed an increase in systems that performed more sophisticated semantic inferences. Among the submissions was the work of Bos & Markert [4], who improved on their RTE-1 results by making several changes to their method. In particular, they added new features to the shallow analysis, such as a measure of the length of T and H , in words. Moreover, they refined their semantic representation of T - H pairs by resolving third-person personal pronouns to named entities and treating proper names and definite descriptions as anaphoric. Lastly, they used more advanced model builders (Paradox 1.3 and Mace 2.0).

However, the authors reported a slight *decrease* in performance when logical inference was combined with word overlap to determine textual entailment. Specifically, the *recall* of their best deep feature (entailment) was very low. This indicates that their logic prover had a very small coverage, which was a problem in their earlier work. It was also the case that proofs were mostly found for examples that had a high word overlap. This made the deeper semantic analysis redundant, since the shallow method already classified these pairs correctly. Hence, the results of Bos & Markert's system were not as positive as they would have liked, given the efforts they made to implement a FOL-based theorem prover.

4.2 Improving Performance with Natural Logic

The failure of Bos & Markert's deep semantic analysis seems to suggest that logical inference is not a worthwhile addition to a system that is designed to recognize textual entailment. However, the work of Chambers et al. [5] casts doubt on this

hypothesis. Their RTE-3 submission was a hybrid of several NLP techniques, including an inference engine that was based on *natural logic* – a system of logical inference that operates over natural language. The authors reasoned that although the coverage of this technique was still rather small, its high precision made it a useful addition to their core RTE system.

Moreover, they argued that inference with natural logic is superior to inference with a FOL-based logic prover, since it avoids the difficulties of translating T and H into first-order logic. To support this claim, they cited the results of Bos & Markert’s RTE-2 submission [4] – they achieved 76% precision on the proofs they found, but their system produced proofs for only 4% of the entire RTE-2 Test set. In contrast, the system of Chambers et al. made positive predictions on 24% of the RTE-3 Test set, with 68% of these being correct. According to the authors, this increase in coverage (*recall*) made the inference component sufficiently valuable, such that their hybrid system attained an absolute accuracy gain of 3.12% over their basic RTE method.

The motivation and principles behind *natural logic* are presented in MacCartney & Manning [11]. They sought to find a middle path between shallow techniques that lack semantic precision and FOL-based theorem provers that are “excessively brittle”. As a result, they produced the NatLog system – a three-stage technique that transforms the Text into the Hypothesis using a sequence of *atomic edits* (generally representing conceptual contractions and expansions, with truth preservation at each step). Eventually, a decision-tree classifier is used to assign each atomic edit an *elementary entailment relation*. If all of these predicted relations are either “equivalent” or “forward”, then the system decides that T entails H . Given the scope of this paper, it is impossible to provide a further account of the theoretical background behind natural logic – for more information, consult the “Related work” section in [11].

5. CONCLUSION AND FUTURE DIRECTIONS

In this paper, I introduced the task of recognizing textual entailment, based on which the PASCAL RTE Challenge was founded. I described the proceedings of this challenge over its three-year existence, with an emphasis on how the competition has matured and grown in popularity, as the number of submissions increased each year. The challenges have shown that a deeper semantic analysis is highly beneficial, if not necessary, for a competitive performance on the RTE task. For this reason, I proceeded to discuss two classes of systems that incorporated logical inference into their design. First, I considered submissions based on First-Order Logic theorem proving. Second, I examined hybrid systems that included an inference component, among other things.

Given this overview of past systems, one might wonder about the future of the PASCAL RTE challenge and the direction of research on recognizing textual entailment.

In Section 3 it was noted that success on the RTE task seems to require a large amount of background knowledge. This was believed to be the case for systems that used logical inference to detect textual entailment. However, there was one RTE-3 submission that achieved good results – 69% accuracy – without the aid of logic provers, while still drawing on a pool of extensive semantic knowledge. The work of Iftene & Balahur-Dobrescu [10] demonstrated that if a hypothesis transformation method was combined with enough background knowledge, one could perform well on the RTE task. In the next Challenge, it is likely that FOL-based theorem proving systems will use the following resources from [10] to generate more world knowledge axioms:

- Database of acronyms to link acronyms (ex. “US”) to their meanings (ex. “United States”)
- Wikipedia to gather additional background knowledge, such as relations between Named Entities, by extracting information about these entities and searching for standard patterns in this description to automatically produce new axioms.
- Extended WordNet to acquire more world knowledge in the form of lexical relations between concepts. This resource has already been incorporated into the work of Tatu & Moldovan [12], with positive results.

However, the system that achieved the best performance on RTE-3 – with an accuracy score of 80% - did not require much training data or background knowledge. Hickl & Bensley [9]’s approach was quite unique and elaborate. They did not incorporate logical inference into their method, though Dagan et al. claimed they did in [7]. Perhaps it is because they used ideas from logic-based methods in their work. For example, they began by extracting a set of publicly held beliefs (*discourse commitments*) from the text-hypothesis pairs, which is similar to what Akhmatova & Molla did in [1], attempting to divide T and H into atomic propositions. Also, at a certain point in their algorithm, Hickl & Bensley used the shallow analysis techniques of Bos & Markert [4] to train a decision tree to detect entailment between a commitment from T and a commitment from H . Regardless of how it was classified, Hickl & Bensley’s approach was rather novel – and successful – which means that it will likely inspire researchers to devise similar algorithms for the next Challenge.

In the future, as the competition intensifies and systems become more successful at recognizing textual entailment, one might wonder about the ultimate goal of this endeavour. In fact, the desired outcome of the challenges has been explicit from the very beginning. It is best articulated by the organizers of RTE-3: “Hopefully, research on textual entailment will finally lead to the development of entailment “engines”, which can

be used as a standard module in many applications (similar to the role of part-of-speech taggers and syntactic parsers in current NLP applications)” [7]. Thus, in just three years, a new research community has formed and has already taken enormous steps towards solving a problem that would not only benefit specific application domains, but the field of Computational Linguistics as a whole.

6. REFERENCES

- [1] E. Akhmatova and D. Molla. (2005). **Recognizing Textual Entailment Via Atomic Propositions**. In *Proceedings of the First Challenge Workshop on Recognizing Textual Entailment*. Southampton, U.K., April 11-13, 2005.
- [2] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. (2006). **The Second PASCAL Recognising Textual Entailment Challenge**. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*. Venice, Italy, April 10, 2006.
- [3] J. Bos and K. Markert. (2005). **Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment**. In *Proceedings of the First Challenge Workshop on Recognizing Textual Entailment*. Southampton, U.K., April 11-13, 2005.
- [4] J. Bos and K. Markert. (2006). **When logical inference helps determining textual entailment (and when it doesn't)**. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*. Venice, Italy, April 10, 2006.
- [5] N. Chambers, D. Cer, T. Grenager, D. Hall, C. Kiddon, B. MacCartney, M. C. de Marneffe, D. Ramage, E. Yeh, and C. D. Manning. (2007). **Learning Alignments and Leveraging Natural Logic**. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*. Prague, Czech Republic, June 28-29, 2007.
- [6] I. Dagan, B. Magnini, and O. Glickman. (2005). **The PASCAL Recognising Textual Entailment Challenge**. In *Proceedings of the First Challenge Workshop on Recognizing Textual Entailment*. Southampton, U.K., April 11-13, 2005.
- [7] I. Dagan, B. Dolan, D. Giampiccolo, and B. Magnini. (2007). **The Third PASCAL Recognizing Textual Entailment Challenge**. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*. Prague, Czech Republic, June 28-29, 2007.
- [8] A. Fowler, B. Hauser, D. Hodges, I. Niles, A. Novischi, and J. Stephan. (2005). **Applying COGEX to Recognize Textual Entailment**. In *Proceedings of the First Challenge Workshop on Recognizing Textual Entailment*. Southampton, U.K., April 11-13, 2005.
- [9] A. Hickl and J. Bensley. (2007). **A Discourse Commitment-Based Framework for Recognizing Textual Entailment**. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*. Prague, Czech Republic, June 28-29, 2007.

- [10] A. Iftene and A. Balahur-Dobrescu. (2007). **Hypothesis Transformation and Semantic Variability Rules Used in Recognizing Textual Entailment**. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*. Prague, Czech Republic, June 28-29, 2007.
- [11] B. MacCartney and C. D. Manning. (2007). **Natural Logic for Textual Inference**. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*. Prague, Czech Republic, June 28-29, 2007.
- [12] M. Tatu and D. Moldovan. (2007). **COGEX at RTE3**. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*. Prague, Czech Republic, June 28-29, 2007.