Rule Chaining and Approximate Match in textual inference

Asher Stern[1], Eyal Shnarch[1], Amnon Lotan[2], Shachar Mirkin[1], Lili Kotlerman[1], Naomi Zeichner[1], Jonathan Berant[2], and Ido Dagan[1]

[1]Bar-Ilan University, Israel
[2]Tel-Aviv University, Israel

October 27, 2010

Abstract

This paper describes the participation of Bar-Ilan university in the sixth RTE challenge. Our textual-entailment engine, BiuTee , was enhanced with new components that introduce chaining of lexical-entailment rules, and tackle the problem of approximately matching the text and the hypothesis after all available knowledge of entailment rules was utilized. We have also re-engineered our system aiming at an open-source open architecture. BiuTee's performance is better than the median of all-submissions, and outperforms significantly an IR-oriented baseline.

1 Introduction

This paper describes Bar-Ilan University's submissions to the Sixth Recognising Textual Entailment (RTE-6) challenge.[1] This year we have focused on enhancing BiuTee (Bar-Ilan University Textual Entailment Engine, [MBHB+09], [BHBD+08], [BHDG+07]) in three main aspects, concerning its software architecture design, its use of knowledge and the entailment classification algorithm.

With respect to its architecture, we have redesigned and reimplemented BiuTee to address two goals: (i) Providing an *open source* public-domain environment which will enable developers of end-applications to embed Textual Entailment (TE) as an underlying semantic module; (ii) Providing an *open architecture* that will enable TE researchers and newcomers to the RTE community to experiment with our system and extend it with new inference components.

Knowledge is a key factor for successful entailment recognition. Yet, RTE experiments by many groups showed inconsistent impact of various entailment-knowledge resources, particularly when different resources are utilized together. We developed a novel graph-based algorithm that integrates lexical entailment relations from multiple knowledge resources, such as WordNet [Fel98] or CatVar [HD03]. The graph represents lexical terms as nodes and entailment relations between them as edges. It enables integrating and chaining entailment relations (rules) obtained from multiple resources. It also enabled us to easily test different integration and filtering policies, in order to yield the most effective recall-precision tradeoffs.

BiuTee determines whether a text entails a given hypothesis primarily through applying a sequence of knowledge-based parse tree transformations over the text. In the vast majority of the cases, remaining gaps between the deduced parse trees and the hypothesis require classifying the pair based on some sort of approximate matching. Data analysis showed that many remaining gaps can be bridged by assuming several relaxations on syntactic structures. Based on that, we developed

[1]`http://www.nist.gov/tac/2010/RTE/index.html`

an algorithm for syntactic-based approximate matching which considers the identity of the main predicate and the importance of various syntactic positions relative to it.

Enhanced with the above three modifications (work in progress), BIUTEE archived a micro-averaged F1 score of 37.5% in the Main task. We compared our approach to a purely-lexical method, relying on an IR-based filtering of the provided candidate list. Our experiments showed that a significant improvement was achieved. BIUTEE was also applied as-is (as tuned for the Main task) to the Novelty Detection and to the Knowledge Base Population tasks, as described in Sections 6.3 and 6.4.

The rest of the paper is organized as follows. Section 2 describes BIUTEE's main principles and its workflow. In Section 3 we present the goals and ideas behind the system's architecture. Section 4 introduces the graph-based approach for combining lexical knowledge and details the other knowledge resources we employed, and in Section 5 we present our new approach for approximate matching. Section 6 includes the results of our submissions and our analysis and in Section 7 we conclude and describe some future work.

2 BIUTEE

The Bar-Ilan University Textual Entailment Engine (BIUTEE) is a transformation-based entailment system making use of various types of entailment knowledge. Knowledge is uniformly represented in the form of *entailment rules* (denoted $LHS \Rightarrow RHS$) allowing consistently applying the same kinds of transformations on the text regardless of the source of the knowledge [BHDGS07]. The applied transformations generate multiple consequents (new texts entailed from the original one), whose parse trees are efficiently stored in a packed representation, termed *Compact Forest* [BHBD09]. The goal is to gradually transform the text such that it becomes more similar to the hypothesis, while maintaining the entailment relationship between the original text and each of its consequents. An approximate matching phase makes the final entailment decision by assessing the degree of syntactic match between the hypothesis and the generated consequents, compensating for knowledge gaps in the available rules. As in RTE-5 [MBHB$^+$09], we require that sentences also pass an IR-based filter, thus combining lexical and syntactic considerations.

BIUTEE, as used in RTE-6, follows the same main principles of the systems we used in previous RTE challenges [BHDG$^+$07, BHBD$^+$08, MBHB$^+$09], yet, this year it has gone through several major enhancements, as described below. First, the system's software architecture was redesigned and BIUTEE was completely reimplemented in order to support an open-architecture and as a major step towards its release as an open source system (see Section 3). Second, the approximate matching component was replaced. Instead of using a supervised learning mechanism, which is highly uninterpretable, a syntax-based matching algorithm is applied. This algorithm considers the hypothesis' main predicate and its main syntactic dependencies, checking whether they are covered by the forest of consequents generated from the text (cf. Section 5). Third, we allowed the chaining of lexical entailment rules via a novel entailment graph algorithm (cf. Section 4).

Overall, the system flow is as follows:

Preprocessing: As a first step the documents are processed by a set of standard NLP tools, including a named entity recognizer [FGM05], the BART coreference resolver [VPP$^+$08] and a number normalizer. Each text sentence is then processed with MINIPAR [Lin98], and represented as a forest of dependency parse trees (initially with a single tree). The forest is augmented with additional information, such as coreference relations with other sentences.

Rule application: Using a set of entailment knowledge resources, entailment rules are applied to generate new consequents for each sentence. These consequents are added to the parse forest

generated for the original sentence [BHBD09]. In this work we applied lexical rules from WordNet [Fel98] and and CatVar [HD03], lexical-syntactic rules from FRED [BASD10], as well as in-house developed generic syntactic rules (a subset of the rules described in [BH10]).

Approximate matching: Mainly due to knowledge gaps, most texts cannot be transformed such that they cover the hypothesis parse tree entirely. To determine whether the remaining gap between the text's deduced forest and the hypothesis is small enough to consider the pair as adhering to entailment, we applied a novel approximate matching algorithm. In contrast to our previous systems, where supervised learning was applied for this purpose [BHBD$^+$08, MBHB$^+$09], in this submission we use a more explicit model, comparing directly the syntactic structures of the hypothesis and the forest of consequents. In comparison to [BHDG$^+$07], where a cost function was used to account for all uncovered components, here we employ some relaxations on the syntactic structure focusing specifically at the coverage of the main predicate and its primary arguments. Section 5 describes this algorithm in detail.

IR-based filtering: In the spirit of the retrieval component of our system from the RTE-5 Search task, we apply a filtering phase to retrieve only a subset of sentences to be considered as candidates for entailment. As can be seen in Section 6.1 this filter improves performance even though an initial filtering was already applied by the challenge organizers in the form of a list of candidate sentences provided for each hypothesis. We index the corpus and search it using the Lucene search engine[2], where each hypothesis constitutes a search query. Sentences from the hypothesis' topic are retrieved and ordered by the rank of their relevance to the query. Among the top-k retrieved sentences, only sentences which are also included in the hypothesis' original candidate list are considered as candidates.

3 Open architecture

In addition to our research efforts, we aim at making BiuTee an open source and open architecture system. This motivation is supported by the following two insights.

First, similar to other NLP infrastructure tasks, textual-entailment is not intended to be an end application. As stated at [DGM05], textual entailment is an attempt to promote an abstract generic task that captures major semantic inference needs across applications. In practice, textual-entailment engines can be embedded in other applications, much like sentence-splitters, parsers and other NLP utilities.

Second, our system, as well as several other systems (e.g. [CH09], [WN09] and [IM09]), uses a large set of tools, resources and algorithms to tackle the textual-entailment challenge. As systems become more complex, the quality of new resources and methods should be measured by their impact on current leading systems, rather than as stand-alone methods. Thus, it is important that newcomers to textual-entailment research will have a comprehensive system available, such that the quality of new ideas and resources will be measured by their contribution to such system.

The flexibility and usability of the BiuTee system are supported by the following properties:

1. All pre-processing tools have a simple and well formed interface, such that replacing or extending them is a relatively easy process.

2. All utilized knowledge resources fit the same formalism, uniformly represented as entailment rules.

[2] `http://lucene.apache.org`

3. The whole system is well designed, well documented and meets open-source standards, such that specific algorithms and methods can be changed relatively easily.

4 Knowledge resources

We employ several knowledge resources to cope with the challenge of semantic variability, aiming to identify the multiple ways in which the same meaning may be expressed. For instance, from reading the phrase "*the IRA announced*" we can infer that there is a "*statement from the Irish Republican Army*" even though they do not share any content words. In order to recognize this, an inference system must know the following entailment rules: *IRA* $\Rightarrow$ *Irish Republican Army* as well as *X announce* $\Rightarrow$ *statement from X*. We use knowledge resources to provide such rules. In an entailment rule the left-hand-side (LHS) entails its right-hand-side (RHS). A rule side may be either a term (a *lexical* rule - the first rule above) or a template which is a parse sub-tree with variables (a *lexical-syntactic* rule - the second rule).

Our largest resource provides lexical entailment rules (entailment relations between terms) and is described next. Sections 4.2 and 4.3 describe the use of lexical-syntactic rules and generic syntactic rules.

4.1 Lexical entailment graph

Our main source for entailment rules, and the one which showed to be the most effective (see Section 6), is a lexical entailment graph. The graph combines various knowledge resources of lexical entailment rules. Terms, corresponding to rule sides, are represented as nodes in this graph and a directed edge between two nodes indicates an entailment relation between the corresponding terms. Given a hypothesis H and a set of lexical entailment rules R, we build the graph G as follows:

1. for each content word h in H add a node to G
2. for each node n in G not yet expanded, expand n:
3. retrieve from R rules whose RHS is n, and for each rule:
4. find its LHS as a node in G or add it as a new node if it does not exist
5. add an edge from the LHS node to n
6. repeat steps $2-5$, for K iterations

By transitivity over graph edges we are able to infer new rules which are not included in any of the input knowledge resource. For example, the result of combining the rule *announce* $\Rightarrow$ *announcement* derived from CatVar with *announcement* $\Rightarrow$ *statement* derived from the WordNet's hypernym relation is *announce* $\Rightarrow$ *statement*. This is the lexical component needed for one of the rules in the example mentioned at the beginning of this section.

The knowledge resources we used were *WordNet* [Fel98] and the *CatVar* (Categorial Variation) database [HD03]. We assigned a sense from WordNet to each node in the graph. At this stage, we do not apply any word sense disambiguation algorithms. Terms which were added from resources with no sense indication were assigned with their first WordNet sense. This known heuristic was found to give better results than considering all senses of a term.

From WordNet we consider synonyms, hyponyms, derivations, verb entailment and meronyms as specifying entailment rules. CatVar is a database of clusters of uninflected words (lexemes) and their categorial (i.e. part-of-speech) variants (e.g. *announce* (verb), *announcer* and *announcement* (noun),

and *announced* (adjective)). The CatVar database contains 63,146 clusters and 109,807 words. We deduce a bi-directional entailment rule between any two lexemes in the same cluster. In future work we aim to incorporate additional knowledge resources into our graph.

After building the graph G for a hypothesis H it can provide the inference system with lexical entailment rules relevant to H. For each term t in a text T and for each h in H, the inference system utilizes G to determine whether entailment holds between t and h. If there is a path from t to h in G, the entailment rule $t \Rightarrow h$ can be used by the system.

In this representation we can easily tune several parameters to reflect precision or recall orientation. For instance, we can choose which knowledge resources to use or limit K, the number of expansion iteration. We found that setting K to 2 maximizes F_1 on the development set.

4.2 Lexical-syntactic rules

Many times transformations over the term level alone do not suffice. Consider the example above: even if we get the rule *announce* $\Rightarrow$ *statement* from the lexical entailment graph, we still need to know that *IRA*, the subject of *announce*, becomes the complement of *statement* in order to infer that *the IRA announced* $\Rightarrow$ *statement from the IRA*. This kind of information can be obtained from resources of lexical-syntactic rules, such as DIRT [LP01], Argument-mapped WordNet [SD09], BInc resource [SD08] and FRED [BASD10]. Most of the available lexical-syntactic resources were extracted from corpora based on statistical techniques and are known to be somewhat noisy. This year we used the potentially more accurate FRED resource[3] of entailment rules learned automatically from the manually constructed resource of FrameNet [BFL98]. FRED is an algorithm which generates unary entailment rules, for templates with one variable, such as *cure X* $\Rightarrow$ *X's recovery*. The resource was reported to yield better performance than WordNet over the ACE Information Extraction dataset[4], while being complementary to it.

In future work we intend to incorporate other resources of lexical-syntactic entailment rules within a lexical-syntactic entailment graph, similar to the lexical graph presented in the previous section, based on directions presented in [BDG10].

4.3 Generic syntactic rules

This resource is composed of two dozen handcrafted generic syntactic-based inference rules, which are a subset of the rules in [BH10, chapter 6]. The rule base captures entailment patterns associated with common syntactic constructs. It is applied iteratively several times on the text in order to produce all possible entailments from each matched construction, such as:

- Apposition: *Ted, the boss, is coming* $\Rightarrow$ *Ted is the boss*
- Conjunction: *Ted has been in meetings and appointments* $\Rightarrow$ *Ted has been in appointments*
- Genitive: *Ted's wife is also coming* $\Rightarrow$ *The wife of Ted is also coming*
- Determiner Alternation: *He's coming to this dinner* $\Rightarrow$ *He's coming to a dinner*
- Relative Clauses: *I'm looking for someone invited by Paula* $\Rightarrow$ *someone was invited by Paula*
- Passive-Active: *He was invited by Paula* $\Rightarrow$ *Paula invited him*

[3]The rule-set is available at: http://www.cs.biu.ac.il/~nlp/downloads/

[4]`http://projects.ldc.upenn.edu/ace/`

5 Syntax based approximate match

5.1 Motivation and Data analysis

In our proof system each proof step adds a new tree to the forest of trees which are entailed from the text. Then, the hypothesis is said to be proven only if at least one of the forest's trees is identical to the hypothesis tree.

Empirical experiments of that method on past years' RTE data-sets showed that complete derivation of H from T is not feasible for most (T, H) pairs. Since such a proof system would classify almost all (T, H) pairs as negative (not entailing), an alternative method of final entailment classification should be defined. The method we used in the past was to collect features which characterize the gap between the forest created by rule applications and the hypothesis and feed them into a machine-learning classifier. The main conceptual drawback of that method is that it is inconsistent with the main idea of the system, which is the derivation of H from T using rule-applications. Moreover, past years' analysis has shown that the main features that had an influence on the final classification were eventually lexical features. An empirical experiment that compared the system to an inference system that was based solely on lexical coverage has shown that the difference between the results of the two systems was only about 1% [MBHB+09], which means that the syntactic structure had a minor effect, if any, on the final results.

Based on the above observations, one of our main research directions is to develop a model that will directly estimate the gap between T and H which cannot be bridged by rule-application based on the given entailment rule-bases. In positive pairs the remaining gap is either because the absence of needed rules, or because the required transformations cannot be defined by deterministic entailment-rules (while in negative pairs the remaining gap is just because T does not entail H).

As a first step we examined how common it is to have a remaining gap that is inherently caused solely by the difference in the syntactic structure between the forest trees and H. To do this we conducted the following experiment: We theoretically assumed the availability of a perfect rule-base, and performed rule-applications manually. Rule applications amounted to performing modifications over the text, assuming perfect lexical, lexical syntactic and syntactic rule-bases, aimed to make the text as close as possible to the hypothesis. The experiment was done on 50 (T, H) pairs from the RTE-5 development data-set, with 25 positive pairs and 25 negative pairs. We tested three phenomena of the resulting inferred tree:

1. Lexical coverage of the hypothesis' content words, measured as the percentage of its words covered after all (manually) imaginable rule applications
2. Coverage of the hypothesis' main verb (we did not require coverage of the verb "to be", since its absence may imply a difference in the tree structure, rather than a lexical gap).
3. Whether the resulting tree is identical to the hypothesis tree.

The lexical gap is examined in the first two phenomena, while the syntactic structure gap is examined in the third one. As shown in Table 1, the results indicate the existence of syntactic-structure gaps in most cases.

5.2 The matching algorithm

5.2.1 Representation

Since our current research direction is in its initial stages, and a principled approximate matching model was not yet developed for the current RTE challenge, we have implemented an intermediate

	positive pairs	negative pairs
Lexical coverage of content words	76%	8%
Coverage of the hypothesis' main verb	84%	48%
The resulting tree is identical to the hypothesis tree	12%	0%

Table 1: manual data analysis

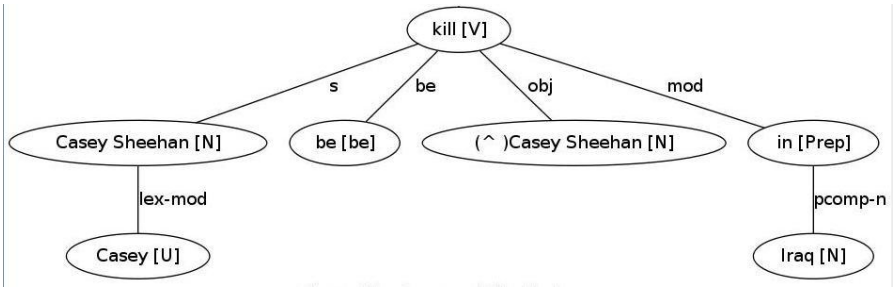

Figure 1: Syntactic dependency tree of "Casey Sheehan was killed in Iraq"

heuristic model that focuses on gaps in syntactic structure. Its main idea is that there are certain syntactic relations whose absence implies that the hypothesis is not entailed from the text, while the absence of other syntactic relations may be ignored. The heuristic we employed was that relations between a verb and its argument must be covered (except for the "be" verb), while other syntactic relations may be ignored. Ignoring syntactic relations in a given sub-tree actually means treating that sub-tree as a bag-of-words.

More formally, a parse tree can be represented in one of two ways. The first representation is the standard *bag-of-words*. This representation is employed when the main verb of the sentence is "be". The second representation, named *verb-argument structure*, issued for all other cases and is a bit more complex, as follows: In this representation a tree is represented by the contents of its root node verb, and the relations between the root and its arguments, where each argument is represented as a bag-of-words. For example, consider the sentence (hypothesis 7 from the Development-set. See parse tree in Figure 1):

Casey Sheehan was killed in Iraq.

The root is "kill", and it has two arguments. One argument is the object "Casey Sheehan" (represented in the parse tree as surface subject, and related to the verb as object). The second argument is "in Iraq". Thus, the tree is represented as

⟨ kill, (object {Casey Sheehan}) (other {Iraq}) ⟩

In our implementation, relations of arguments to the root were limited to the following three types: *subject*, *object* and *other*. Also note that a bag-of-word contains only the content words of the corresponding sub-tree.

Similarly, consider the following sentence (sentence 7 of document APW_ENG_20050807.0002 from the Development set. See parse tree in Figure 2):

Her son, Casey, 24, was killed in Sadr City, Iraq, on April 4, 2004.

The representation of that sentence is

⟨kill (object {Son, Her, Casey, 24}) (other {Sadr City, Sadr, Iraq, April 4 2004, April, 4})⟩

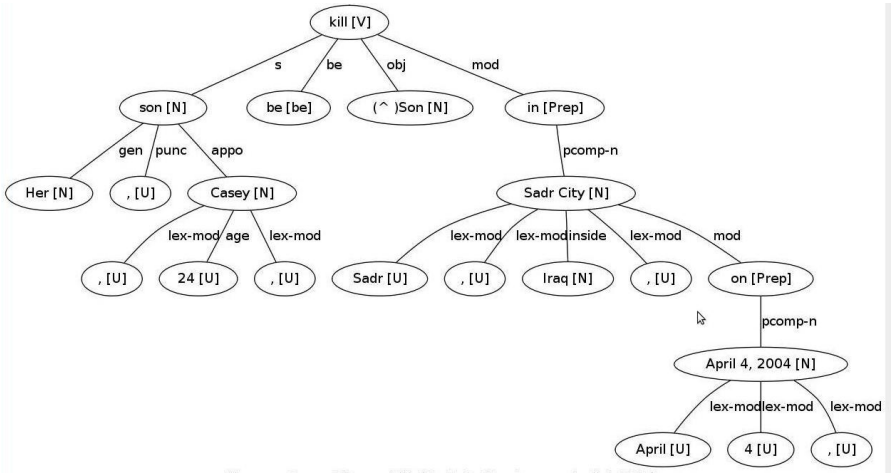

Figure 2: Syntactic dependency tree of "Her son, Casey, 24, was killed in Sadr City, Iraq, on April 4, 2004"

5.2.2 Matching criterion

The system classifies a (T, H) pair as *entailing*, if and only if at least one of the forest's trees *matches* the hypothesis tree, where *match* is defined as follows:

For trees that are represented as *bag-of-words*, a forest tree matches the hypothesis tree if the percentage of the words coverage reaches a predefined threshold.

For trees that are represented as *verb-argument structure*, a forest tree matches the hypothesis tree if the verbs in their roots are identical, and the percentage of covered arguments reaches a predefined threshold. An argument in the hypothesis tree is defined as covered by an argument from the forest tree if their relations to the verb are identical, and the percentage of covered words in the argument's bag-of-words reaches a predefined threshold.

In conclusion, three thresholds are defined:

1. Threshold of word coverage for trees that are represented as *bag of words*
2. Threshold of the percentage of covered arguments
3. Threshold of word coverage within an argument

Theses three thresholds were tuned on the development-set.

6 Results

6.1 Main task

Our best result ("BIU1") achieved a micro-averaged F1 score of 37.50% on the Test-set. To find out how our system performs comparing to a baseline, we adopted the baseline that was defined for the RTE-5 pilot search task. In that baseline, each hypothesis is used as a query to the Lucene IR system, which returns a list of scored sentences for each query (hypothesis). The baseline classifies the top ranked K sentences for each hypothesis as entailing, and classifies the others as non-entailing. We found that the baseline's F1 measure is 30.4% on the test-set (setting $K = 11$, which is the best K found for the development-set), about 7% lower than our system.

We submitted three runs configured as described in Table 2. In the first and the third runs we used the IR phase (see Section 1), setting the value of K to 28, which was found as optimal

Run #	Coreference resolution	Knowledge resources	IR filtering
BIU1	BART	Lexical entailment graph (WordNet + CatVar) Syntactic rules	Top 28
BIU2	BART	Lexical entailment graph (WordNet + CatVar) Syntactic rules	None
BIU3	BART	Lexical entailment graph (WordNet + CatVar) Syntactic rules Lexical syntactic rules: FRED	Top 28

Table 2: Main & Novelty task system configurations

BIUTEE RTE-6 results						
	Development set			Test set		
Run #	Recall %	Precision %	F1 %	Recall %	Precision %	F1 %
BIU1	36.01	46.88	**40.73**	37.46	37.54	**37.50**
BIU2	34.00	39.87	36.70	36.08	29.40	32.40
BIU3	38.68	41.71	40.14	37.88	33.36	35.48
General statistics over all submissions of RTE-6						
Highest result						48.01
Lowest result						11.60
Median over all results						33.72

Table 3: Main task results

on the development-set. The first and second runs differ only in the activation of the IR phase. The third run is similar to the first, except that a different knowledge configuration was used. The micro-averaged results for the Main task are summarized in table 3.

6.2 Ablation tests

We submitted three ablation tests, all of them relative to the first run, "BIU1". In the first ablation test we evaluate the contribution of WordNet and in the second the contribution of CatVar. In the third we evaluate the contribution of BART as a coreference resolution system. The ablation tests results are summarized in table 4. The first two ablation tests show consistent positive contribution of both WordNet and CatVar as resources for entailment rules. The third ablation test shows a consistent negative effect of the coreference resolution system.

6.3 Novelty Detection task

The Novelty Detection subtask is based on the Main Task and is aimed at specifically addressing the interests of the Summarization community, in particular with regard to the Update Summarization

Tested (eliminated) component	dev set		test set	
	F1 %	Δ %	F1 %	Δ %
WordNet	39.18	+1.55	36.60	+0.90
CatVar	40.20	+0.53	36.87	+0.63
BART (coreference resolution)	41.62	−0.89	38.38	−0.88

Table 4: Ablation tests

	Primary score			Secondary score		
run #	Recall %	Precision %	F1 %	Recall %	Precision %	F1 %
BIU1	75.00	73.53	74.26	36.38	34.83	35.59
BIU2	70.00	71.43	70.71	34.58	26.94	30.28
BIU3	67.00	77.91	72.04	38.04	29.76	33.39

Table 5: Novelty task results

task. The structure of the novelty detection task is identical to the main task's structure. Both tasks use the same document-corpus, but have a different set of hypotheses and candidate entailing sentences. In the novelty detection task, given a hypothesis H, if any sentence from the text-corpus entails H then H is considered (in the gold standard) as not *novel*, while if H cannot be entailed from the corpus then it is considered as *novel*. Therefore, in addition to the score used for the main task (named "secondary score", or "justification score"), a new score (named "primary score", or "novelty score") evaluates the answers *for each hypothesis* in the following way: If the system found at least one entailing sentence for a given H then it means that the system classified H as non-novel. In contrast, if no entailing sentences were detected then it means that the system classified H as novel.

We applied our Main task's system on the novelty detection task as-is, without any parameter tuning. Experiments on the Main task showed that parameter tuning can improve results by up to 5%, but due to time constraints we did not perform tuning for the novelty detection task. Table 5 summarizes our system's results on the Novelty task test set.

6.4 Knowledge Base Population task

In the Knowledge Base Population task (KBP) the data set consists of 23,192 pairs of the form *(T,* $\mathbb{H}$*)*, where T is a document and $\mathbb{H}$ is a set of about 3-6 *semantically equivalent* hypotheses related to it. Thus, the objective in this task is to decide whether each T entails (all or none of) the hypotheses in $\mathbb{H}$. To accomplish this, our system initially determines the entailment between each T and every member of the $\mathbb{H}$ set separately, just as it would in the other tasks. Subsequently, if more than a quarter of the hypotheses in each pair are found to be entailed, the system returns a positive answer for that pair. Otherwise, the answer is negative.

Tackling this task was a far greater technical challenge than the other two. First, because the sheer size of the data set demanded much longer run time. Second, since the test set was unexpectedly much harder to process than the development set, featuring many Ts longer, by two orders of magnitude, than those in the development-set. Consequently, our team did not have time to fine tune BiuTee's parameters for this task, and it was run without using the Co-Reference engine, for speed. The best result was obtained by the BIU2 submission, with 16% F1 micro-averaged on the test set (39.5% recall and 10% precision).

7 Conclusions

In this paper we presented two main components for improving the Bar-Ilan university inference system, BiuTee. The first is a graph structure for representing lexical-entailment rules, implemented as a dynamic data structure which facilitates the integration and chaining of different lexical entailment resources. The second component is a novel method to approximately match the remaining gap between the text and the hypothesis, which could not be bridged by applying entailment rules from our various knowledge resources.

For the current RTE challenge we have implemented only preliminary algorithms along our current research lines, in which we intend to develop more principled methods to be introduced in the future. Comparing to all last year's systems in the RTE-5 Search Task, which were outperformed by the Lucene top-K baseline, our system's result is about 7% over that baseline, demonstrating the contribution of various components.

Our system participated, in addition to the main task, in two new tasks: Novelty detection task, and KBP validation task. Those tasks demonstrate the potential contribution of the Textual-Entailment paradigm to other NLP tasks.

Several ideas for future work were mentioned in the paper. One idea is developing a principled model for estimating the remaining gap between the forest of consequents inferred from the text and the hypothesis. Ideas related to knowledge resources include incorporating knowledge from additional resources (e.g. Wikipedia) into the lexical entailment graph, providing each lexical rule with a score or probability, and incorporating other resources of lexical-syntactic entailment rules within a lexical-syntactic entailment graph, based on directions presented in [BDG10].

8 Acknowledgements

This work was partially supported by the Negev Consortium of the Israeli Ministry of Industry, Trade and Labor, the PASCAL-2 Network of Excellence of the European Community FP7-ICT-2007-1-216886, the FIRB-Israel research project N. RBIN045PXH and the Israel Science Foundation grant 1112/08. Jonathan Berant is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

References

[BASD10] Roni Ben Aharon, Idan Szpektor, and Ido Dagan. Generating entailment rules from framenet. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 241–246, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[BDG10] Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of focused entailment graphs. In *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229, Morristown, NJ, USA, 2010. Association for Computational Linguistics.

[BFL98] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *IN PROCEEDINGS OF THE COLING-ACL*, pages 86–90, 1998.

[BH10] Roy Bar-Haim. *Semantic Inference at the Lexical-Syntactic Level.* PhD thesis, Bar-Ilan Univeristy, 2010.

[BHBD+08] Roy Bar-Haim, Jonathan Berant, Ido Dagan, Iddo Greental, Shachar Mirkin, Eyal Shnarch, and Idan Szpektor. Efficient semantic deduction and approximate matching over compact parse forests. In *Proceedings of Text Analysis Conference (TAC)*, 2008.

[BHBD09] Roy Bar-Haim, Jonathan Berant, and Ido Dagan. A compact forest for scalable inference over entailment and paraphrase rules. In *Proceedings of EMNLP*, 2009.

[BHDG+07] Roy Bar-Haim, Ido Dagan, Iddo Greental, Idan Szpektor, and Moshe Friedman. Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

[BHDGS07] Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. Semantic inference at the lexical-syntactic level. In *AAAI*, pages 871–870. AAAI Press, 2007.

[CH09] Peter Clark and Phil Harrison. An inference-based approach to recognizing entailment. Proceedings of the First Text Analysis Conference, Gaithersburg, Maryland, United States, National Institute of Standards and Technology (NIST), 2009.

[DGM05] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *MLCW*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer, 2005.

[Fel98] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.

[FGM05] Jenny R. Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, 2005.

[HD03] Nizar Habash and Bonnie Dorr. A categorial variation database for english. In *Proceedings of the North American Association for Computational Linguistics*, pages 96–102, Edmonton, Canada, 2003.

[IM09] Adrian Iftene and Mihai-Alex Moruz. Uaic participation at rte5. In *Proceedings of TAC*, Gaithersburg, Maryland, 2009.

[Lin98] Dekang Lin. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC 1998*, Granada, Spain, 1998.

[LP01] Dekang Lin and Patrick Pantel. Dirt @sbt@discovery of inference rules from text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328, New York, NY, USA, 2001. ACM.

[MBHB+09] Shachar Mirkin, Roy Bar-Haim, Jonathan Berant, Ido Dagan, Eyal Shnarch, Asher Stern, and Idan Szpektor. Addressing discourse and document structure in the RTE Search Task. In *Proceedings of TAC*, Gaithersburg, Maryland, 2009.

[SD08] Idan Szpektor and Ido Dagan. Learning entailment rules for unary templates. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 849–856, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[SD09] Idan Szpektor and Ido Dagan. Augmenting wordnet-based inference with argument mapping. In *TextInfer '09: Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 27–35, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[VPP+08] Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Ro Moschitti. Bart: A modular toolkit for coreference resolution. In *In Association for Computational Linguistics (ACL) Demo Session*, 2008.

[WN09] Rui Wang and Guenter Neumann. An accuracy-oriented divide-and-conquer strategy for recognizing textual entailment. In *Proceedings of TAC*, Gaithersburg, Maryland, 2009.