Addressing Discourse and Document Structure in the RTE Search Task

Shachar Mirkin[§], **Roy Bar-Haim**[§], **Jonathan Berant**[†],
Ido Dagan[§], **Eyal Shnarch**[§], **Asher Stern**[§], **Idan Szpektor**[§]

§ Computer Science Department, Bar-Ilan University, Ramat-Gan 52900, Israel
† The Blavatnik School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract

This paper describes Bar-Ilan University's submissions to RTE-5. This year we focused on the Search pilot, enhancing our entailment system to address two main issues introduced by this new setting: scalability and, primarily, document-level discourse. Our system achieved the highest score on the Search task amongst participating groups, and proposes first steps towards addressing this challenging setting.

1 Introduction

Bar-Ilan's research focused this year on the Search task, which brings about new challenges to entailment systems. In this work we put two new aspects of the task in the center of attention, namely scalability and discourse-based inference, and enhanced our core textual entailment engine to address them.

While the RTE-5 search dataset is relatively small, we aim at a scalable system that can search for entailing texts over large corpora. To that end, we apply as a first step a retrieval component which considers each hypothesis as a query, expanding its terms using lexical entailment resources. Only sentences with sufficiently high lexical match are retrieved and considered as candidate entailing texts, to be classified by the entailment engine, thus saving the system a considerable amount of processing.

In the Search task, sentences are situated within a set of documents. They rely on other sentences for their interpretation and their entailment is therefore dependent on other sentences as well. Hence, discourse and document-level information play a crucial role in the inference process (Bentivogli et al., 2009). In this work we identified several types of discourse phenomena which occur in a discourse-dependent setting and are relevant for inference. As existing tools for coreference and discourse processing provide only limited solutions for such phenomena, we suggest methods to address their gaps. In particular, we examined complementary methods to identify coreferring phrases as well as some types of bridging relations which are realized in the form of "global information" perceived as known for entire documents. As a first step, we considered phrase pairs with a certain degree of lexical overlap as potentially coreferring, but only if no semantic incompatibility is found between them. For instance, noun phrases which have the same head, but their modifiers are antonyms, are ruled out. We addressed the issue of global information by identifying and weighting prominent document terms and allowing their inference even when they are not explicitly mentioned in a sentence. To account for coherence-related discourse phenomena – such as the tendency of entailing sentences to be adjacent to each other – we apply a two-phase classification scheme, where a second-phase meta-classifier is applied, extracting features that consider the initial independent classification of each sentence.

Using the above ideas and methods, our system obtained a micro-averaged F_1 score of 44.59% on the Search task.

The rest of this paper is organized as follows. In Section 2 we describe the core system used for running both the Main and the Search tasks, highlighting the differences relative to the system used in our RTE-4 submission. In Section 3 we describe the Main task's submission. In Section 4 we present our approach to the Search task, followed by a description of the retrieval module (Section 5) and the way we address discourse aspects of the task (Section 6). Description of the submitted systems and their results are detailed in Section 7. Section 8 contains conclusions and suggestions for future work.

2 The BIUTEE System

For both the Main and Search tasks we used the Bar-Ilan University Textual Entailment Engine (BIUTEE), based on the system used for our RTE-4 submission (Bar-Haim et al., 2008). BIUTEE applies transformations over the text parse-tree using a knowledge-base of diverse types of entailment rules. These transformations generate many consequents (new texts entailed from the original one), whose parse trees are efficiently stored in a packed representation, termed *Compact Forest* (Bar-Haim et al., 2009). A classifier then makes the entailment decision by assessing the coverage of the hypothesis by the generated consequents, compensating for knowledge gaps in the available rules.

The following changes were applied to BIUTEE in comparison with (Bar-Haim et al., 2008): (a) several syntactic features are added to our classification module, as described below; (b) a component for supplementing coreference relations is added (see Section 6.1); (c) a different set of entailment resources is employed, based on performance measured on the development set.

Further enhancements of the system for accommodating to the Search task are described in Sec. 6.

Knowledge resources. A variety of knowledge resources may be employed to induce parse-tree transformations, as long as the knowledge can be represented as entailment rules (denoted *LHS* ⇒ *RHS*). In our submissions, the following resources for entailment rules were utilized. See Sections 3 and 7 for the specific subsets of resources used in each run:

- Syntactic rules: These rules capture entailment inferences associated with common syntactic constructs, such as conjunction, relative clause, apposition, etc. (Bar-Haim et al., 2007).
- *WordNet* (Fellbaum, 1998): The following WordNet 3.0 relationships were used: synonymy, hyponymy (two levels away from the original term), the hyponym-instance relation and derivation.
- *Wiki*: All rules from the Wikipedia-based resource (Shnarch et al., 2009) with DICE co-occurrence score above 0.01.
- *DIRT*: The DIRT algorithm (Lin and Pantel, 2001) learns entailment rules between binary predicates, e.g. *X explain to Y* ⇒ *X talk to Y*. We used the version described in (Szpektor and Dagan, 2007), which learns canonical rule forms applied over the Reuters Corpus, Volume 1 (RCV1)[1].

The above resources are identical to the ones used in our RTE-4 submission. For RTE-5 we also used the following sources of entailment-rules:

- *Snow*: Snow et al.'s (2006) extension to WordNet 2.1 with 400,000 additional nodes.
- *XWN*: A resource based on Extended WordNet (Moldovan and Rus, 2001), as described in (Mirkin et al., 2009).
- A geographic resource, denoted *Geo*, based on TREC's TIPSTER gazetteer. We created "meronymy" entailment rules such that each location entails the location entities in which it is found. For instance, a city entails the county, the state, the country and the continent in which it is located, and a country entails its continent. To attend to ambiguity of location names, often polysemous with common nouns, this resource was applied only when the candidate geographic name in the text was identified as representing a location by the Stanford Named Entity Recognizer (Finkel et al., 2005).
- *Abbr*: A resource containing about 2000 rules for abbreviations, where the abbreviation entails the complete phrase (e.g *MSG* ⇒ *Monosodium Glutamate*). Rules in this resource were generated based on the abbreviation lists of BADC[2] and Acronym-Guide[3].

Lastly, we added a small set of rules we developed addressing lexical variability involving temporal phrases. These rules are based on regular expressions and are generated on the fly. For example, the occurrence of the date 31/1/1948 in the text triggers the generation of a set of entailment rules including: *31/1/1948* ⇒ {*31/1, January, January 1948, 20th century, forties*} etc. We refer to this resource as *DateRG* (*Date Rule Generator*).

Classification. BIUTEE's classification component is based on a set of lexical and lexical-syntactic features, as described in (Bar-Haim et al., 2008). Analysis of those features showed that the lexical

[1] http://trec.nist.gov/data/reuters/reuters.html

[2] http://badc.nerc.ac.uk/help/abbrevs.html

[3] http://www.acronym-guide.com/

Run	Accuracy (%)
Main-BIU$_1$	63.00
Main-BIU$_2$	**63.80**

Table 1: Results of our runs on the Main task's test set.

features have the most significant impact on the classifier's decision. Thus, we have engineered an additional set of lexical-syntactic features:

1. A binary feature checking if the main predicate of the hypothesis is covered by the text. The main predicate is found by choosing the predicate node closest to the parse-tree root. If no node is labeled as a predicate, we choose the content-word node closest to the root. On the development set this method correctly identifies the main predicate of the hypothesis in approximately 95% of the cases.
2. Features measuring the match between the subject and the object of the hypothesis' main predicate and the corresponding predicate's arguments in the text.
3. A feature measuring the proportion of NP-heads in the hypothesis that are covered by the text.

3 The Main Task

We submitted two runs for the 2-way Main task, denoted *Main-BIU*$_1$ and *Main-BIU*$_2$. *Main-BIU*$_1$ uses the following resources for entailment rules: the syntactic rules resource, *WordNet*, *Wiki*, *DIRT*, *Geo*, *Abbr* and *DateRG*. *Main-BIU*$_2$ contains the same set of resources with the exception of *Geo*. Table 1 details the accuracy results achieved by our system on the Main task's test set.

Table 2 shows the results of ablation tests relative to *Main-BIU*$_1$. As evident from the table, the only resource that clearly provides leverage is *WordNet*, though performance was also improved by using *DIRT*. These two results are consistent with our previous ones (Bar-Haim et al., 2008), while *Wiki*, which was helpful in previous work, was not in this case. Further analysis is required to determine the reason for performance degradation by this and other resources. The two preliminary resources that handle abbreviations and temporal phrases did not provide any marginal contribution over the other resources and are therefore excluded from the table.

Resource removed	Accuracy (%)	ΔAccuracy (%)
WordNet	60.50	2.50
DIRT	61.67	1.33
Geo	63.80	-0.80
Wiki	64.00	-1.00

Table 2: Results of ablation tests relative to *Main-BIU*$_1$. The columns from left to right specify, respectively, the name of the resource removed in each ablation test, the accuracy achieved without it and the marginal contribution of the resource. Negative figures indicate that the removal of the resource increased the system's accuracy.

4 Addressing the Search Task

The pilot Search task presents new challenges to inference systems. In this task, an inference system is required to identify all sentences that entail a certain hypothesis in a given (small) corpus. In comparison to previous RTE challenges, the task is closer to practical application setting and better corresponds to natural distribution of entailing texts in a corpus.

The task may seem at first look as a variant of Information Retrieval (IR), as it requires finding specific texts in a large corpus. Yet, it is fundamentally different from IR for two main reasons. First, the target output is a set of sentences, each one of them evaluated independently, rather than a set of documents. Consequently, a system has to handle target texts which are not self-contained, but are rather dependent on their surrounding text. Hence, discourse is a crucial factor. Second, the decision criterion is *entailment* rather than *relevancy*.

A naïve approach may be applied to the task by reducing it to a set of text-hypothesis pairs and applying Main-task techniques on each pair. However, as evident from the development set, where entailing sentences account for merely 4% of the sentences[4], such an approach is highly inefficient, and might not be feasible for larger corpora. Note that only limited processing of test sentences can be done in advance, while most of the computational effort is required at inference time, i.e. when the sentence is assessed for entailment of a specific given hypothesis. Hence, we chose to address the Search task with an approach in the spirit of IR (passage retrieval) for Question Answering (e.g. (Tellex et al., 2003)):

First, we apply a simple and fast method to filter the sentences based on lexical coverage of the hypothesis in each sentence, discarding from fur-

[4] 810 out of over 20,000 possible sentence-hypothesis pairs.

ther processing any document in which no relevant sentences are found. Such a filter reduces significantly the amount of sentences that require deeper processing, while allowing tradeoff between precision and recall, as required. Next, we process and enrich non-filtered sentences with discourse and document-level information. These sentences are then classified by a set of supervised classifiers, based on features extracted for each sentence independently. Meta-features are then extracted at the document-level based on the output of the aforementioned classifiers, and a meta-classifier is applied to determine the final classification.

The details of our retrieval module, the implementation for addressing discourse issues and the two-tier classification process are described in the next two Sections.

5 Candidate Retrieval

The retrieval module of our system is employed to identify candidates for entailment: For each hypothesis h, it retrieves candidate sentences based on their term coverage of h. A word w_h in h is *covered* by a word in a sentence s, w_s, if they are either identical (in terms of their stems[5]) or if a lexical entailment rule $w_s \Rightarrow w_h$ is found in the currently employed resource-set. A sentence s is retrieved for a hypothesis h if its coverage of h (percentage of covered content words in h) is equal or is greater than a certain predefined threshold. The threshold is set empirically by tuning it over the development set for each set of resources employed.

At preprocessing, each sentence in the test set corpus is tokenized, stemmed and stop-words are removed. Given an hypothesis it is processed the same way. We then utilize lexical resources to apply entailment-based expansion of the hypothesis' content words in order to obtain higher coverage by the corpus sentences and consequently – a higher recall. For example, the following sentence covers three out of the six content words of the hypothesis simply by means of (stemmed) word identity:

h : "*Spain took steps to legalize homosexual marriages*"

s : "*Spain's Prime Minister . . . made legalising gay marriages a key element of his social policy.*"

Using WordNet's synonymy rule *gay* $\Leftrightarrow$ *homosexual* increases the coverage from $\frac{1}{2}$ to $\frac{2}{3}$.

This retrieval process can be performed within minutes on the entire development or test set, with any set of the resources we employed.

6 Discourse Aspects of the Search Task

As mentioned, discourse aspects play a key role in the Search task. We therefore analyzed a sample of the development set's sentence-hypothesis pairs, looking for discourse phenomena that are involved in the inference process. In the following subsections we describe the prominent discourse and document-structure phenomena we have identified and addressed in our implementation. These phenomena are typically poorly addressed by available reference resolvers and discourse processing tools, or fall completely out of their scope.

6.1 Non-conflicting coreference matching

A large number of coreference relations in our sample are comprised of terms which share lexical elements, such as *the airliners's first flight* and *the Airbus A380's first flight*. Although common in coreference relations, it turns out that standard coreference resolution tools miss many of these cases.

For the purpose of identifying additional coreferring terms, we consider two noun phrases in the same document as coreferring if: (i) their heads are identical and (ii) no semantic incompatibility is found between their modifiers. The types of incompatibility we handle in our current implementation are antonymy and mismatching numbers. For example, two nodes of the noun *distance* would be considered incompatible if one is modified by *short* and the second by *long*. Similarly, two nodes for *dollars* are considered incompatible if they are modified by different numbers. By allowing such lenient matches we compensate for missing coreference relations, potentially resulting in an increased overall system recall. The precision of this method may be further improved by adding more types of constraints to discard incompatible pairs. For example, it can be verified that modifiers are not co-hyponyms (e.g. *dog food*, *cat food*) or otherwise semantically disjoint. These additional coreference relationships are augmented to each document prior to the classification stage.

[5] For stemming we used the Porter Stemmer from: http://www.tartarus.org/~martin/PorterStemmer

6.2 Global information

Key terms or prominent pieces of information that appear in the document, typically at the title or the first few sentences, are many times perceived as "globally" known throughout the document. For example, the geographic location of the document's theme, mentioned at the beginning of the document, is assumed to be known from that point on, and will often not be mentioned in further sentences which do refer to that location.

This is a bridging phenomenon that is typically not addressed by available discourse processing tools. To compensate for that, we implemented the following simple method: We identify key terms for each document based on *TF-IDF* scores, requiring a minimum number of occurrences of the term in the document and giving additional weight to terms in the title. The top-n ranking terms are considered *global* for that document. Then, each sentence parse tree in the document is augmented by adding the documents' global terms as nodes directly attached to the sentence's root node. Thus, an occurrence of a global term in the hypothesis is matched in each of the sentences in the document, regardless of whether the term explicitly appears in the sentence. For example, global terms for the topic discussing the ice melting in the Arctic, typically contain a location such as *Arctic* or *Antarctica* and terms referring to *ice*, like *permafrost*, *icecap* or *iceshelf*.

Another method for addressing missing coreference relations is based on the assumption that adjacent sentences often refer to the same entities and events. Thus, when given a sentence for classification, we also consider the text of its preceding sentence. Specifically, when extracting classification features for a given sentence, in addition to the features extracted from the parse tree of the sentence itself, we extract the same set of features[6] from the joint tree composed of the tree representations of the current and previous sentences put together.

6.3 Document-level classification

Beyond discourse references addressed above, further information concerning discourse and document structure phenomena is available in the Search setting and may contribute to entailment classification. For example, we observed that entailing sentences tend to come in bulks. This reflects a common coherence aspect, where the discussion of a specific topic is typically continuous rather than scattered across the entire document, and is especially apparent in long documents. This *locality* phenomena may be useful for entailment classification since knowing that a sentence entails the hypothesis increases the probability that adjacent sentences entail the hypothesis as well. More generally, for the classification of a given sentence, useful information can be derived from the classification results of other sentences in the document, reflecting other discourse and document-level phenomena.

To that end, we use a meta-classification scheme with a two-phase classification process, where a *meta-classifier* utilizes entailment classifications of the first classification phase to extract *meta-features* and determine the final classification decision. This scheme also provides a convenient way to combine scores from multiple classifiers used in the first classification phase. We refer to these as *base-classifiers*. This scheme and the meta-features we used are detailed hereunder.

Let us write (s, h) for a sentence-hypothesis pair. We denote the (set of pairs in the) development (training) set as $\mathcal{D}$ and in the test set as $\mathcal{T}$. We split $\mathcal{D}$ into two halves, $\mathcal{D}_1$ and $\mathcal{D}_2$. We rely on document-level information to determine entailment. Thus, for a given h, following the candidate retrieval stage, we process all pairs corresponding to h paired with each sentence in the documents containing the candidates. These additional pairs are not considered as entailment candidates and are always classified as non-entailing. We write $\mathcal{R}$ for the set of candidate pairs and $\mathcal{R}'$ for the set containing both candidates and the abovementioned additional pairs. Note that $\mathcal{R} \subseteq \mathcal{R}' \subseteq \mathcal{T}$. We make use of n base-classifiers, $C_1, \ldots, C_n$, among which $C^\star$ is a designated classifier with additional roles in the process, as described below. Classifiers may differ, for example, in their classification algorithm. An additional meta-classifier is denoted C_M.

The classification scheme is shown as Algorithm 1. We now elaborate on each of these steps.

At Step 1, features are extracted for every (s, h) pair in the training set by each of the base-classifiers. These include the same features as in the Main task, as well as the features for the joint forest of the current and previous sentence described in

[6] Excluding the tree-kernel feature in (Bar-Haim et al., 2008)

```
Training
1: Extract features for every $(s, h)$ in $\mathcal{D}$
2: Train $C_1, \ldots, C_n$ on $\mathcal{D}_1$
3: Classify $\mathcal{D}_2$, using $C_1, \ldots, C_n$
4: Extract meta-features for $\mathcal{D}_2$ using the
   classification of $C_1, \ldots, C_n$
5: Train $C_M$ on $\mathcal{D}_2$

Classification
6: Extract features for every $(s, h)$ in $\mathcal{R}'$
7: Classify $\mathcal{R}'$ using $C_1, \ldots, C_n$
8: Extract meta-features for $\mathcal{R}$
9: Classify $\mathcal{R}$ using $C_M$
```

Algorithm 1: Meta-classification

Section 6.2. In steps 2 and 3, we split the training set into two halves (taking half of each topic), train n different classifiers on the first half and then classify the second half of the training set using each of the n classifiers. Given the classification scores of the n base-classifiers to the (s, h) pairs in the second half of the training set, $\mathcal{D}_2$, we add in Step 4 the following meta-features to each pair:

- **Classification scores:** The classification score of each of the n base-classifiers. This allows the meta-classifier to integrate the decisions made by different classifiers.
- **Second-closest entailment:** Considering the locality phenomenon described above, we add as feature the distance to the second-closest entailing sentence in the document (including the sentence itself), according to the classification of $C^\star$. Formally, let i be the index of the current sentence and J be the set of indices of entailing sentences in the document according to $C^\star$. For each $j \in J$ we calculate $d_{i,j} = |i - j|$, and choose the second smallest $d_{i,j}$ as d_i. If entailing sentences indeed always come in bulks, then $d_i = 1$ for all entailing sentences, but $d_i > 1$ for all non-entailing sentences.

  Let us further explain the rationale behind this score: Suppose we compute the distance to the *closest* entailing sentence rather than to the *second-closest* one. Thus, it is natural to assume that we do not count the sentence as closest to itself since it disregards the environment of the sentence altogether, eliminating the desired effect. If $C^\star$ mistakenly classifies a sentence as entailing, but all sentences in its environment are not entailing, both scheme of the closest entailing sentence (excluding self) and the second-closest (including self) produce the same distance. On the other hand, under the 'closest' scheme, both an entailing sentence at the "edge" of an entailment bulk and the non-entailing sentence just next to it, have a distance of 1: suppose that sentences $i, \ldots, i + l$ constitute a bulk of entailing sentences. Then:

  $d_{i-1} = |(i - 1) - i| = 1$ and
  $d_i = |i - (i + 1)| = 1$

  Under our scheme, however, the non-entailing sentence has a distance of 2 while the entailing sentence has a distance of 1, since we consider both the sentence's own classification and its environment's classification. We scale the distance and add the feature score: $-\log(d_i)$.
- **Smoothed entailment:** This feature also addressed the locality phenomenon by smoothing the classification score of sentence i with the scores of adjacent sentences, weighted by their distance from the current sentence i. Let $s(i)$ be the score assigned by $C^\star$ to sentence i. We add the Smoothed Entailment feature score:

  $$\text{SE}(i) = \frac{\sum_w (b^{|w|} \cdot s(i + w))}{\sum_w (b^{|w|})} \qquad (1)$$

  where $0 < b < 1$ is a parameter and w is an integer bounded between $-N$ and N, denoting the distance from sentence i.
- **1^{st} sentence entailing title:** As shown in (Bensley and Hickl, 2008), the first sentence in a news article typically entails the article's title. We found this phenomenon to hold for the RTE-5 development set as well. We therefore assume that for each document $s_1 \Rightarrow s_0$ where s_1 and s_0 are the document's first sentence and title respectively. Hence, under entailment transitivity, if $s_0 \Rightarrow h$ then $s_1 \Rightarrow h$. The corresponding binary feature states whether the sentence being classified is the first sentence of the document *AND* the title entails the hypothesis according to $C^\star$.
- **Title entailment:** In many texts, and in news articles in particular, the title and the first few sentences are often used to present the entire document's content and may therefore be considered as a summary of the document. Thus, it

may be useful to know whether these sentences entail the hypothesis, as an indicator to the general potential of the document to include entailing sentences. Two binary features are added according to the classification of $C^{\star}$ indicating whether the title entails the hypothesis and whether the first sentence entails it.

After adding the meta-features we train a meta-classifier on this new set of features in Step 5. Test sentences that passed the retrieval module's filtering then go through the same process: features are extracted for them and they are classified by the already trained n classifiers (Steps 6 and 7), meta-features are extracted in Step 8, and a final classification decision is performed by the meta-classifier in Step 9.

7 Search Task - Experiments and Results

We submitted three distinct runs for the Search task, as described below.

***Search-BIU*$_1$** Our first run determines entailment between a sentence s and a hypothesis h purely based on term coverage of h by s, i.e. by using the retrieval module's output directly (cf. Section 5). For picking the best resource-threshold combination for candidate retrieval, we assessed the performance of various settings for term expansion. These include the use of *WordNet*, *Wiki*, *XWN*, and Dekang Lin's distributional similarity resource (Lin, 1998), as well as unions of these resources and the basic setting where no expansions at all are used. Each expansion setting was assessed with a threshold range of 10%-80% on the development set. Several such settings are are shown in Table 3. As seen in the Table, the best performing setting in terms of micro-averaged F_1 – which is therefore used for *Search-BIU*$_1$ – was the use of *Wiki* with a 50% coverage threshold, achieving a slightly better score than using no resources at all.

***Search-BIU*$_2$** In this run BIUTEE is used, in its standard configuration, i.e., a single classifier is used and features are extracted for each sentence independently, without attending to document-level considerations. Test-set sentences are pre-filtered by the retrieval module using no resources for expansion[7] and with minimum 50% coverage of the hypothesis. The entailment resources used in this run are: syntactic rules, *WordNet*, *Wiki*, *Geo*, *XWN*, *Abbr*, *Snow* and *DateRG*. For the classifier, we use the *SVM*^{perf} package (Joachims, 2006) with a linear kernel. Global information is added by enriching each sentence with the top-three terms from the document, based on the *TF-IDF* scores (cf. Section 6.2), if they occur at least three times in the document, while title terms are counted twice.

Resource	Min. Coverage	P (%)	R (%)	F_1(%)
Wiki	50%	35.5	42.8	**38.8**
-	50%	**35.6**	41.5	38.3
XWN	50%	30.5	**46.2**	36.7
WordNet	60%	30.8	43.6	36.1
WN+Wiki	60%	30.3	43.8	35.8
Lin	80%	22.9	35.2	27.7

Table 3: Performance of lexical resources for expansion on the development set showing the best coverage threshold found for each resource when using the retrieval module to determine entailment. Note that settings using different thresholds are not directly comparable.

***Search-BIU*$_3$** Here, our complete system is applied, using the meta-classifier, as described in Section 6.3. The retrieval module's configuration and the set of employed entailment resources are identical to the ones used in *Search-BIU*$_2$. In this system, we used two base-classifiers ($n = 2$): *SVM*^{perf} and Naïve Bayes from the WEKA package (Witten and Frank, 2005), where the first among these is set as our designated classifier $C^{\star}$ which is used for the computation of the document-level features. *SVM*^{perf} was also used for the meta-classifier. For the smoothed entailment score (cf. Section 6.3), we used $b = 0.9$ and $N = 3$, based on tuning on the development set.

The results obtained in each of the above runs are detailed in Table 4. For easier comparison we also show the results of another lexical run, termed *Search-BIU*$_{1'}$, where no expansion resources are used, as in *Search-BIU*$_2$ and *Search-BIU*$_3$. Hence, *Search-BIU*$_{1'}$ can be directly viewed as the candidate retrieval step of the next two runs. The entailment engine in these two runs applies a second filter to the candidates based on the inference classification results, aiming to improve precision of this initial set. Recall is, therefore, limited by that of the candidate retrieval step.

Although achieving rather close F1 scores, we note that our submissions' outputs are substantially

[7] We picked this configuration empirically . Note that systems may have different optimal retrieval configurations.

Run	P (%)	R (%)	F1 (%)
Search-BIU$_1$	37.03	**55.50**	44.42
Search-BIU$_{1'}$	37.15	53.50	43.85
Search-BIU$_2$	40.49	47.88	43.87
Search-BIU$_3$	**40.98**	51.38	**45.59**

Table 4: Micro-average results of our Search task runs.

different from each other, as reflected in the number of sentences classified as entailing: while *Search-BIU*$_1$ marked 1199 sentences as entailing (1152 for *Search-BIU*$_{1'}$), in *Search-BIU*$_2$ and *Search-BIU*$_3$ the numbers are 946 and 1003, respectively. Comparing *Search-BIU*$_{1'}$ to *Search-BIU*$_3$ based on Table 4 and these figures, we learn that 149 sentences are removed by the latter, of which 89% are false-positives. This directly translates to a 10% relative increase in precision with an approximate 4% relative recall loss. We further learn by comparing *Search-BIU*$_2$ and *Search-BIU*$_3$ that the meta-classification scheme – constituting the difference between the two systems – is helpful, mainly for recall increase. Which ones of the meta-features are responsible for the improved performance requires further analysis.

An interesting observation concerning the datasets is obtained by comparing the second line in each of Tables 3 and 4, referring to lexical runs with no expansions, which retrieve sentences based on direct matches between the sentence and hypothesis terms. On the test set, this configuration achieves a recall score higher by 29% relatively to the recall obtained on the development set (53.5% vs. 41.5%), with an even slightly higher precision. Apparently, the test set was much more prone to favor lexical methods than the development set. This may contribute to understanding why our complete system achieved only little leverage over the purely lexical run. In any case, it constitutes a bias between the datasets, significantly affecting systems' training and tuning.

We refrained from performing analysis on the Search task's test set as we intend to perform further experiments using this dataset.

8 Conclusions and Future Work

In this work we addressed the RTE Search task, identified key issues of the task, and have put initial solutions in place. We designed a scalable system in which we addressed various document-structure and discourse-based phenomena which are relevant for inference under such settings. A thorough analysis is required to understand the impact of each of our system's components and resources. So is the development of sound algorithms for addressing the discourse phenomena we pointed out.

Our system achieved the highest score among the groups that participated in the challenge, but has surpassed our own baseline by only a small margin. Previous work, e.g. (Roth and Sammons, 2007; Adams et al., 2007) showed that lexical methods constitute a strong baseline for RTE systems. Our own results provide another support for this observation. Still, by applying our inference engine, we were able to improve precision relative to the lexical system, thus improving the overall performance in terms of F_1. This constitutes a way to tradeoff recall and precision depending on one's needs. We believe that further improvement can be achieved by recruiting IR and QA know-how to the retrieval phase and by providing more comprehensive implementations for the ideas we proposed in this paper.

Acknowledgments

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

Rod Adams, Gabriel Nicolae, Cristina Nicolae, and Sanda Harabagiu. 2007. Textual entailment through extended lexical overlap and lexico-semantic matching. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. 2007. Semantic inference at the lexical-syntactic level. In *AAAI*.

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

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

Jeremy Bensley and Andrew Hickl. 2008. Unsupervised resource creation for textual inference applications. In Bente Maegaard Joseph Mariani Jan Odjik Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of LREC*.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, Medea Lo Leggio, and Bernardo Magnini. 2009. Considering discourse references in textual entailment annotation. In *Proceedings of the 5th International Conference on Generative Approaches to the Lexicon (GL2009)*.

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

Jenny R. Finkel, Trond Grenager, and Christopher Manning. 2005. 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.

Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*.

Shachar Mirkin, Ido Dagan, and Eyal Shnarch. 2009. Evaluating the inferential utility of lexical-semantic resources. In *Proceedings of EACL*, Athens, Greece.

Dan Moldovan and Vasile Rus. 2001. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of ACL*.

Dan Roth and Mark Sammons. 2007. Semantic and logical inference model for textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Eyal Shnarch, Libby Barak, and Ido Dagan. 2009. Extracting lexical reference rules from Wikipedia. In *Proceedings of ACL-IJCNLP*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of COLING-ACL*.

Idan Szpektor and Ido Dagan. 2007. Learning canonical forms of entailment rules. In *Proceedings of RANLP*.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of SIGIR*.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, San Francisco.