Recognizing Textual Entailment With a Modified BLEU Algorithm

Abstract

The BLEU algorithm was proposed as a baseline technique for the task of recognizing textual entailment (RTE) by (Pérez & Alfonseca, 2005) in the first PASCAL RTE challenge. However, because the BLEU algorithm was designed as a metric for measuring the accuracy of automatically generated translations, certain features of the algorithm are not appropriate for RTE. Specifically, BLEU penalizes brevity both explicitly and implicitly in its scoring algorithm; since entailment hypothesis are very often short phrases, this behavior is not desirable in RTE. Therefore, this paper proposes a variation on the BLEU scoring algorithm that does not penalize brevity and can consistently outperform both the unmodified BLEU algorithm and a dumb baseline.

1 Introduction

The task of recognizing textual entailment is a difficult one that can be approached in a number of ways. The first PASCAL Challenge for Recognizing Textual Entailment (RTE1) received submissions that applied various semantic, syntactic, and statistical methods towards performing this task (Dagan et al. 2005), all of which were based on sound theoretical footing, but none of which achieved eye-catching results.

Within the framework of the RTE1 challenge, a dumb baseline of 50% could be achieved by simply labeling all entailments as *true* (or as *false*): in comparison, the highest accuracy achieved by any RTE1 submission was 58.6%. The modest algorithm outlined in this paper achieves an accuracy of 53.8% on the same test set.

The fact that this modest approach outperforms 5 of the 16 original RTE1 submissions (and performs comparably to several of the others), most of which involve much more complicated systems, is indicative of the current plight of RTE. As with any discipline in its early stages, simple systems can offer initial strong results, while the more intricate systems will require further development before their potential becomes apparent (Bayer et al. 2005). (See my other paper for the promising approaches for RTE)

Despite the fact that this paper illustrates that the BLEU algorithm can be adjusted to perform better on RTE, I do not think this is an area that warrants further work. I can imagine certain potential applications for the BLEU algorithm within RTE (e.g. as a metric for comparing the hypothesis H to atomic propositions generated by a RTE system), but the algorithm is not useful on its own as anything other than a baseline.

1.1 Recognizing Textual Entailment

Before going too far, it will be helpful to briefly outline what exactly is meant by textual entailment. For a more thorough discussion of the textual entailment task and definition, see (Dagan et al. 2005, sections 1-2).

In a nutshell, textual entailment is loosely defined to hold if the meaning of a hypothesis text H can be inferred by an average human reader (using only his knowledge of English and some general world knowledge) given a text T. For example, below are two T-H pairs. In the first, entailment holds (the entailment is judged to be *true*) while in the second, entailment does not hold (it is judged *false*)

T: Satomi Mitarai died of blood loss.
H: Satomi Mitarai bled to death.
(*True*)

T: Coyote shot after biting girl in Vanier Park.

H: Girl shot in park.
(*False*)

Many entailments are mere paraphrases,[1] while some are more involved and require the application of world knowledge. Entailment is said to be applicable to a number of different fields within NLP (see section 3.2 for a list of the subfields recognized by RTE1), where it is desirable to know whether the information in a hypothesis can be derived from a given source text.

The fact that so many entailments are paraphrases, repeating some or all of T in H, is the main reason that the BLEU algorithm works for RTE. As will be seen, n-grams cannot hope to capture every case of entailment, but the BLEU algorithm can tailored to suit the RTE task.

2 The Algorithm

2.1 The Original BLEU Algorithm

The BLEU algorithm was created by (Papineni et al. 2002) as a method for judging the performance of machine translation systems. In this use, BLEU compares the output of a MT system (called the test or hypothesis) to one or more human-generated translations (the reference). The score of the system translation is based on the number of n-grams (with values of n that typically cover the range from 1-4) appearing in the test that also appear in the reference, modified by a brevity factor that penalizes the test for being shorter than the reference (on the fair assumption that any two translations should be roughly equivalent in length).

The scoring algorithm goes something as follows:

1. For each i up to N, calculate a score s_i that is the ratio of the count of i-grams co-appearing in both reference and test ($c_{test,ref}$) and the count of i-grams appearing in the test (c_{test}):

$$s_i = \frac{c_{test,ref}}{c_{test}}$$

2. Average the values of s_i. This is accomplished with a weighted geometric mean; the weight w_i is typically kept constant for all i (w_i=1/N for all i).

$$s_N = e^{\sum_{i=1}^{N} w_i \log(s_i)}$$

3. Calculate the brevity penalty. If the length of the test (t) is greater than the length of the reference (r), then there is no penalty (b=1). Otherwise, the penalty is logarithmically derived from the ratio of the two lengths:

$$b = \begin{cases} e^{(1-r/t)} & \text{if } t < r \\ 1 & \text{if } t > r \end{cases}$$

4. Finally, calculate the overall score as the mean of all scores multiplied by the brevity penalty.

$$s_{bleu} = bs_N$$

Alternatively, all together:

$$s_{bleu} = e^{(1-r/t)} e^{\sum_{i=1}^{N} w_i \log(s_i)}$$

The scoring algorithm described above is that found in Papineni's Perl implementation of BLEU, which was used for the BLEU evaluations used in this study and served as the model for the modified BLEU algorithm described below.

2.2 Modifying the BLEU Algorithm for RTE

Application of the BLEU algorithm to the RTE task makes sense on the level that an entailed hypothesis will very likely (though not necessarily) contain many of the same words that appear in the source text. Thus, the basic core of the algorithm, matching the co-occurrence of n-grams, remains valid.

However, the relationship between text and hypothesis in RTE is not the same as the relationship between test and reference in MT. The most important difference is that while in MT both test and reference are expected to convey the *same* information, in RTE the hypothesis is only expected to contain a *subset* of the information contained in the text. While it is true that in some cases of entailment H will contain roughly the same information

[1] Up to 94% in the RTE1 data. (Bayer et al. 2005)

as T, it is counter to the definition of entailment that H could contain more information than is stated (explicitly or implicitly) in T.

There are several consequences that derive from this fundamental difference. The first consequence is that it is clear which of T and H should be considered the text and which the reference in the BLEU framework. Clearly, the hypothesis should be considered the test (the equivalent of the candidate translation), since we want to count the number of n-grams in H that also appear in T, and not vice-versa.

A further consequence of this difference is that there is no longer a motivation for the brevity penalty. Entailment hypotheses are very often shorter than the source text, by virtue of the fact that they contain only a subset of information in the source text. Thus, directly penalizing the hypothesis for being shorter than the text is not productive in RTE.

It is a simple matter to eliminate the brevity penalty in the BLEU algorithm, but there is actually a second penalty against brief or truncated hypotheses hidden in the scoring algorithm. This arises from the use of a weighted geometric mean to average the *n*-gram scores. Although stated earlier in log terms, the formula for calculating s_n can be equivalently stated as:

$$s_N = \prod_{i=1}^{N} s_i^{w_i}$$

This formulation makes it is easier to see that if s_i is null for any value of i, then the entire score will also be null. This is extremely harsh in the RTE task because often, due to the fact that the hypothesis is a highly summarized or truncated version of the source text, there will be no n-gram overlap for higher values of *n*. For example, 63% of the entailment pairs in the RTE1 development set had no *n*-gram overlap for n=4.[2]

To rectify this problem (clearly we don't want 63% of our data to have a null score), there are two options: we could use a lower value of *n*, or we could change the averaging function. It is not desirable to reduce the value of *n*: 37% is still a significant number of entailments that make use of the 4-gram overlap, and it is likely that these longer phrases represent the algorithm's best hope for capturing syntactic features. Besides, even at n=2, a significant number (15%) of the RTE1 development set would receive a null score. It is preferable to have a continuum of gradated scores than to break the data into essentially null and non-null categories.

The obvious solution is to use a linear, rather than geometric mean. In fact, Papineni et al. state that this same averaging method yielded good results during the development of the BLEU algorithm, but was later discarded because it did not account for the exponential decay in n-gram overlap for increasing values of n. This is less of a concern in RTE, where the main objective of this algorithm is to measure word overlap. Thus, we can use a linear weighted average such as:

$$s_N = \sum_{i=1}^{N} w_i s_i$$

In fact, this average score will act as the overall score in our modified algorithm, since there is no brevity factor. The values for s_i and w_i are calculated as above.

3 Results and Performance

3.1 Evaluation Methodology

The evaluation for the performance of the algorithms was the same as that defined for the RTE1 challenge. Accuracy was measured as the fraction of correctly labeled *true* or *false* entailments as produced by the system (i.e. the percentage of judgments that are correct). A second measure, a confidence-weighted score (cws), was computed. This measure weights judgments based on their relative ranking as follows:

$$cws = \frac{1}{n} \sum_{i=1}^{n} \frac{\#\text{correct up to } i}{i}$$

Although other measures for evaluation, such as precision, recall, and f, have been recognized as potentially insightful for RTE (Dagan et al. 2005), they were not included in this project because they were not part of the RTE1 challenge.

[2] See the section below on unmodified BLEU results for further statistics.

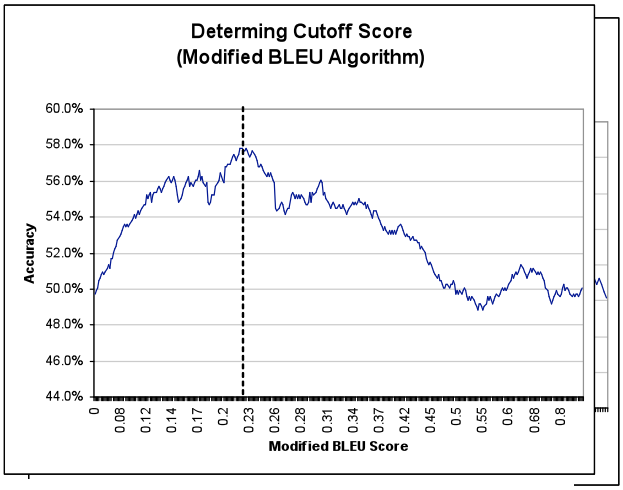

Fig 3: Chart plotting accuracy vs. cutoff score for the RTE1 test data. The dotted line indicates the cutoff score that was chosen for the modified algorithm based on the development set.

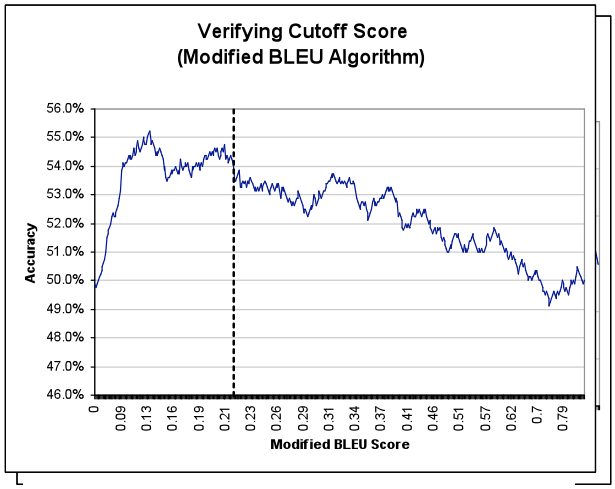

Fig 4: Chart plotting accuracy vs. cutoff score for the RTE1 test data. The dotted line indicates the cutoff score that was chosen based on the development set.

3.2 Datasets

The RTE1 challenge consisted of a development dataset, containing 283 *true* entailments and 284 *false* entailments (567 total entailments), and a test set containing 400 *true* and 400 *false* entailments (800 total entailments). In addition, the RTE2 development set consists of 400 *yes* and 400 *no* entailments (800 total entailments; they decided to change the terminology, but *yes*/*no* denotes the same thing that *true*/*false* denoted in RTE1). Thus, the dumbest baseline for all of these systems, choosing always either *true* or *false*, is 50%.

The examples for the RTE1 datasets were subdivided into categories: these categories correspond to different areas of NLP and the entailments are representative of the type of texts that arise in that area. The categories in RTE1 were: information retrieval (IR); comparable documents (CD); reading comprehension (RC); question answering (QA); information extraction (IE); machine translation (MT); and paraphrase acquisition (PP). Most systems showed slight differences in performance between these categories.

3.3 Results for the Unmodified BLEU Algorithm

The first run used the unmodified BLEU algorithm to assign a score to each text-hypothesis pair in the RTE1 development set. For this run and all others, N was chosen to be 4, which is also the default value in the Perl implementation. After running the algorithm on all T-H pairs, a cutoff score s_{cut} was determined such that all entailments with a score equal to or lower than s_{cut} were judged false, and all entailments with a score higher than s_{cut} were judged true.

As mentioned above, the unmodified BLEU algorithm assigns a high number of zero scores on the RTE dataset. In fact, zero is an attractive cutoff point; as figure 1 above helps illustrate, the optimal accuracy on the development set was found to occur at two points: $s_{cut} = 0.002$ and $s_{cut} = 0.132$. Both of these cutoffs give accuracies of 53.8% (299/567). The lower value was chosen, as it is unclear what is represented by the second peak; a cutoff score of zero, on the other hand, has a certain aesthetic appeal.

At this cutoff, the unmodified BLEU algorithm correctly identified 253 *false* entailments and 46 *true* entailments, for an overall accuracy of 53.8% (299/567). The difference in number of correct *false* and correct *true* judgments is simply due to the fact that the majority of hypotheses received a zero score, thus falling below the cutoff score.

Applying this same cutoff value to the RTE1 test data, the unmodified BLEU algorithm correctly identified 253 *false* entailments and 163 *true* entailments, for an overall accuracy of 52.0% (416/800).

Both figures 1 and 2 illustrate that the unmodified BLEU algorithm achieves its peak accuracy with a cutoff of zero, perhaps because this very low score is most likely to capture the *false* entailments with very low correlation to their source texts. However, it is difficult to explain why the algorithm seems to have an alternative peak at higher cutoff scores: this may be due to the algorithm capturing the very highly correlated *true* entailments (perhaps from the CD or MT tasks). The distributions also seem to indicate a difference in the nature of the development and the test data sets: the potential peak cutoff at 0.12 in the development data would have an awful performance on the test set.

It is probably wise not to read too much into these distributions, as this system is performing at a level only slightly higher than random guessing. The best way to think about the unmodified BLEU algorithm as a RTE system is that it is a binary function: a zero score predicts a *false* entailment, while a nonzero score predicts a *true* entailment.

3.4 Results for the Modified BLEU Algorithm

Since it was specifically altered to give fewer null scores, the modified BLEU algorithm provided a distinctly different distribution of scores than the original BLEU algorithm. It also achieved higher accuracy for the RTE1 development and test data

As before, we chose the optimal cutoff score s_{cut} based on the performance of the algorithm on the RTE1 development set. The ideal value was determined to be $s_{cut} = 0.221$; this corresponds to the maximum accuracy, as can be seen in figure 3. At this value of s_{cut}, the modified BLEU algorithm achieved 57.8% accuracy (328/567), correctly identifying 131 *false* entailments and 197 *true* entailments.

The system did not fare as well on the test data: using the previously determined cutoff score, the modified BLEU algorithm managed only an accuracy of 53.8% (430/800), correctly identifying 160 *false* entailments and 270 *true* entailments.

As a side note, if the algorithm were allowed to re-train based on the test data, the best possible accuracy would be 55.3%, using a s_{cut} of 0.1333; see figure 4. This may be an argument in favor of lowering the cutoff score for future applications.

3.5 RTE2 Datasets

Considering that there seems to have been a significant difference in the make-up of the two datasets in RTE1, it is worthwhile to investigate the performance of these two systems on a new dataset: the development set for the Second Recognizing Textual Entailment challenge.

The RTE2 dataset is slightly different in format consists of the categories IE, IR and QA from above, as well as text summarization (SUM). The RTE2 data were chosen "to provide more 'realistic' text-hypothesis examples, based mostly on outputs of actual systems" (Bar-Haim 2005), and thus one may expect that the datasets will be different in some ways from the RTE1 datasets.

Using the same cutoff score of $s_{cut} = 0.221$, the modified BLEU algorithm achieved an accuracy of 60.4% (483/800) on the RTE2 development data set. Thus, whatever changes the developers made to the datasets seem to favor the *n*-gram approach.

Likewise, the unmodified BLEU algorithm was tested on the RTE2 development set, achieving 56.0% accuracy (448/800) using a cutoff score of zero.

The RTE2 results at least partially validate the choice of cutoff score for the modified BLEU algorithm assigned by the RTE1 development data (0.221). The revised cutoff score based only on the RTE2 development data would be 0.2580, yielding an accuracy of 61.4%, only 1% higher than the accuracy achieved with the original cutoff score. Looking over the results of the RTE1 development and test data combined with the RTE2 development data, it seems best to keep the cutoff score near 0.22.

3.6 Comparison of Results

At this point it seems appropriate to bring up the work of Pérez & Alfonseca from the first RTE challenge. Their submission consisted of an implementation of the BLEU algorithm, but I have been unable to replicate their exact results. From their pseudo-code it would seem that they implemented a variation of the BLEU algorithm that used a weighted linear mean to average the scores. However, they do not mention that they have modified the BLEU algorithm from the version proposed by Papineni et al., nor do they give any of the details of their implementation.

I experimented with a modified version of the algorithm that included the brevity factor but used the linear rather than geometric mean, hoping to match their results. I was unable to come up with the cutoff scores they reported, although I managed to obtain loosely similar accuracy values.

Below I summarize the results of Pérez & Alfonseca's BLEU implementation, my unmodified BLEU implementation, and the modified BLEU implementation. The table contains the accuracy values for these three systems, as well as the best system in RTE1 for comparison:

	BLEU	Modified BLEU	Pérez & Alfonseca	RTE1 Best
Development Set (RTE1)	53.8	57.8	54	n/a
Test Set (RTE1)	52.0	53.8	49.5	58.6
Development Set (RTE2)	56.0	60.4	n/a	n/a

Table 2: Comparison of accuracy values for three BLEU-based systems and the best entry in RTE1.

While the table above illustrates that the modified BLEU algorithm proposed in this study outperforms the unmodified BLEU algorithm and the BLEU variant implemented by Pérez & Alfonseca, it also points out that none of the BLEU-based systems achieve accuracies close to the best system in RTE1, and this gap is not likely to be closed. In the next section, I will discuss the role of BLEU as a baseline in RTE and look at some other promising approaches to the RTE task.

4 Trees and Rocket Ships

(Bayer et al. 2005), in their submission to the RTE1 challenge, rightly point out that RTE is a difficult task and that until the complex systems are able to get their many components working well together the simple systems will outperform them. They warn against trying to "climb a tree to get to the moon" (quoting Dagan). In this sense, the BLEU algorithm is a tree; it gets us part of the way towards the solution, but inevitably leads to a dead end. It may have a role within a larger system, but the future of RTE is such that a simplistic n-gram approach will not be successful on its own.

4.1 Shortcomings of the BLEU Algorithm (the "Tree")

The BLEU algorithm is not meant to be anything more than a baseline for RTE, thus it is not productive to spend much time pointing out its deficiencies. An example or two will suffice to show why an n-gram model will always fail on certain types of entailment pairs.

For example, consider the case where the entire hypothesis H appears as a clause in the text T. The BLEU algorithm will assign this a score of 1, since every possible n-gram in H also appears in T. However, this does not ensure entailment. Consider the pair:

T: It is not the case that John likes ice cream.
H: John likes ice cream.
(*false*)

Our first reaction might be to question the somewhat arbitrary relationship we came up with earlier, treating H as the test and T as the reference in the BLEU algorithm. However, changing this assignment accomplishes nothing, since the labels T and H can similarly be reversed in the above example to yield another false entailment that achieves a BLEU score of 1.

While problems such as this clearly show that BLEU is deficient as a stand-alone algorithm for RTE, it has potential applications as a moderate baseline and possibly as a final stage in more complex systems. One could imagine, for instance, a system that generates atomic propositions from T and uses the BLEU algorithm to compare these propositions to H.

4.2 Other Promising Approaches to RTE (the "Rocket Ships")

Reading through the proceedings of the first RTE challenge workshop, it is clear that there are several interesting approaches being taken towards the RTE task. Below I list a few of my favorites, giving only a general sketch of how they work.

Tree-Edit Distance and Syntactic Graph Matching

I group these together because even though there are some basic differences, they share a similar concept. Both approaches use dependency-tree structures to represent the T and H sentences (or clauses), and then use some procedures to calculate the difference between T and H in terms of the cost required to transform one tree/graph into the other.

Minimum tree-edit distance algorithms use the basic transformations of inserting, removing, and substituting nodes within the trees to find the shortest edit path that transforms the H tree into the T tree (or vice versa). It would be interesting to see a minimum tree-edit distance algorithm for RTE that could incorporate syntactic concepts into its cost calculations: e.g. inserting a "be" verb node into apposition structures to transform them into sentences. This would be a big project in itself! However, I think that there is a lot of room for innovation in this approach and it may offer significant progress in the future. See (Kouylekov & Magnini, 2005), (Raina et al. 2005), and others.

Statistical Lexical Relationships

This approach treats entailment as translation, drawing on concepts from the field of MT. T-H pairs in the training set are aligned using software such as Giza++, and "translate" the text into the entailment. (Bayer et al. 2005)'s System 2 is exactly this type of system. Even though they describe it as a "tree" in their analogy above, it achieved the best performance in the RTE1 challenge!

While these systems may fail to directly address the underlying semantic and syntactic principles that define entailment, there is certainly room for improvement, especially in the view of those who hope to merge syntactic considerations with the statistical methods that have such success in MT. See (Bayer et al. 2005), (Glickman et al. 2005), and others.

Atomic Propositions

This is an interesting approach that basically predicts the entailment hypothesis from the source text. A sentence contains several atomic propositions that each could generate an independent sentence; these atomic propositions are compared to the hypothesis to see if any of them matches. See (Akhmatova, 2005).

Semantic Distance

This is a concept (not really an approach) that is an essential feature of any effective RTE system. There are many types of relatedness between words (synonymy, antynomy, hypernymy, etc.) and a RTE system must be able to use these relationships effectively when performing various comparisons and transformations between T and H. See (Budanitsky et al. 2001) and practically any of the RTE1 submissions.

5 Conclusion

In this paper I hope to have shown that the BLEU algorithm is more effective in dealing with the RTE task when it has undergone certain modifications, namely eliminating the brevity penalty and adjusting the scoring mechanism to use a weighted linear, rather than geometric, mean. The modifications to the system led to a 2-4% increase in accuracy over all test sets when compared to the unmodified BLEU algorithm.

Despite these modifications, the BLEU algorithm remains nothing more than a baseline for recognizing textual entailment, because it lacks the room to grow and accommodate further improvements. This is not to say that it is useless for the future of RTE; it may potentially serve as a baseline for evaluating other systems and components in RTE, and it may itself act as a component in a robust RTE system.

6 Acknowledgements

This study made use of the Perl implementation of the BLEU algorithm by Kishore Papineni (version 4/12/2002).

References

Elena Akhmatova, 2005. *Textual Entailment Resolution via Atomic Propositions*. Proceedings of the First PASCAL Recognizing Textual Entailment Challenge.

Roy Bar-Haim. 2005. *Second Recognizing Textual Entailment Challenge*. http://www.pascal-network.org/Challenges/RTE2/

Samuel Bayer, John Burger, Lisa Ferro, John Henderson, Alexander Yeh. 2005. *MITRE's Submissions to the EU Pascal RTE Challenge*. Proceedings of the First PASCAL Recognizing Textual Entailment Challenge.

Alexander Budanitsky and Graeme Hirst. 2001. *Semantic Distance in WordNet: An Experimental, application-oriented evaluation of five measures.* Workshop on WordNet and Other Lexical Resources.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. *The PASCAL Recognizing Textual Entailment Challenge*. Proceedings of the First PASCAL Recognizing Textual Entailment Challenge.

Oren Glickman, Ido Dagan and Moshe Koppel, 2005. *Web Based Probabilistic Textual Entailment.* Proceedings of the First PASCAL Recognizing Textual Entailment Challenge.

Milen Kouylekov and Bernardo Magnini, 2005. *Recognizing Textual Entailment with Tree Edit Distance Algorithms* Proceedings of the First PASCAL Recognizing Textual Entailment Challenge.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *BLEU: A Method for Automatic Evaluation of Machine Translation*. Proceedings of the 40th Annual Meeting of the ACL.

Diana Pérez and Enrique Alfonseca. 2005. *Application of the BLEU Algorithm for Recognizing Textual Entailments*. Proceedings of the First PASCAL Recognizing Textual Entailment Challenge.

Rajat Raina, Aria Haghighi, Christopher Cox, Jenny Finkel, Jeff Michels, Kristina Toutanova, Bill MacCartney, Marie-Catherine de Marneffe, Christopher D. Manning, Andrew Y. Ng, 2005. *Robust Textual Inference using Diverse Knowledge Sources*. Proceedings of the First PASCAL Recognizing Textual Entailment Challenge.