FBK Participation in RTE6: Main and KBP Validation Task

Milen Kouylekov[1], Yashar Mehdad[1,2], Matteo Negri[1], Elena Cabrio[1,2]
[1]Fondazione Bruno Kessler (FBK-irst), [2]University of Trento
{kouylekov,mehdad,negri,cabrio}@fbk.eu

Abstract

This paper overviews FBK's participation in the *Main* and *KBP Validation Pilot* task organized within the RTE6 Evaluation Campaign. Our submissions have been produced running the EDITS (Edit Distance Textual Entailment Suite) open source RTE package, which allows to experiment with different combinations of algorithms, entailment rules, and optimization strategies. The evaluation on test data confirmed their effectiveness, with good results in both the tasks. Our best run in the Main task achieved a Micro-Averaged F-measure of 44.71% (with the best and the median system respectively achieving 48.01% and 33.72%); our best run in the KBP Validation task achieved the highest score, with 25.5% F-measure.

1 Introduction

As in previous editions of the RTE Campaign, our submissions for RTE6 have been produced using EDITS (Edit Distance Textual Entailment Suite), the open-source software package for recognizing Textual Entailment developed by FBK. The package, which is freely available at *http://edits.fbk.eu/*[1], provides a basic framework for a distance-based approach to the task, with a highly configurable and customizable environment to experiment with different algorithms.

As thoroughly described in [1], the approach implemented by EDITS assumes that the distance between *T* and *H* is a characteristic that separates the positive *T*-*H* pairs, for which the entailment relation holds, from the negative pairs, for which the entailment relation does not hold. More specifically, EDITS is based on edit distance algorithms, and computes the *T*-*H* distance as the overall cost of the edit operations (*i.e.* insertion, deletion and substitution) that are necessary to transform *T* into *H*. For this purpose, EDITS can be easily configured by defining its three basic modules:

- The **edit distance algorithm**, which calculates the set of edit operations that transform *T* into *H*.

[1]The current release of the package, EDITS 2.1, is available under GNU Lesser General Public Licence - LGPL.

- The **cost scheme**, which defines the cost associated to each edit operation involving an element of *T* and an element of *H*.
- Optional sets of **entailment/contradiction rules**, that provide specific knowledge (*e.g.* lexical, syntactic, semantic) about the allowed transformations between portions of *T* and *H*.

Each module, and its corresponding parameters, can be configured by the user through the EDITS Configuration File (ECF). A basic configuration file includes at least one distance algorithm and one cost scheme, while rule repositories can be optional. EDITS provides a general framework which allows, through the ECF, to combine in different ways the existing algorithms/cost schemes, or replace them with new ones implemented by the user.

To allow replicability of the results, and as an additional contribution to the RTE community, the final configurations used for our submissions are freely downloadable from the EDITS website.

The following sections overview our participation in the two tasks, providing details about the experiments carried out at the training stage, the submitted runs, the results achieved, and some additional tests performed to check the validity of what we learned during training.

2 Main task

Given a corpus *C*, a hypothesis *H*, and a set of "candidate" entailing sentences for that *H* retrieved from *C* by the Lucene search engine, the RTE6 main task consists in identifying all the sentences that entail *H* among the candidate sentences.

2.1 Training the system

As a first step in the training stage we created a set of entailment pairs of the type $TCand_x$-*H* for each hypothesis *H* and for each candidate sentence for that *H*. The resulting entailment corpus was then processed using the Stanford Parser [2] to obtain tokenization, lemmatization, part-of-speech tagging, and dependency tree representations of both *T*s and *H*s. Finally, to perform reliable training/evaluation routines, the pre-processed data were splitted into two portions, respectively used for training (*i.e.* learning a distance model from a set of annotated examples) and development (*i.e.* evaluating the learned model over a set of unseen examples). To this aim we experimented with two strategies, namely: *i) random split* (*i.e.* a random separation of training data in two portions of the same size), and *ii) topic split* (*i.e.* a separation based on the *topics* to which documents in the corpus *C* are associated). While the former solution represents a standard way to avoid overfitting problems, the latter solution has the additional purpose of checking if a subset of topics for which the system returns higher and more stable performance can be used to train better models.

Training/evaluation routines have been carried out over pre-processed data, using different configurations of the EDITS system. In particular, building on the lessons learned from previous participations in RTE Campaigns, we experimented with: *i)*

the use of lexical knowledge (in the form of entailment rules mined from different resources including Wikipedia, VerbOcean [3], WordNet [4], Lin's Proximity and Dependency thesauri[5]), *ii)* different performance optimization criteria, *iii)* the use of algorithms working at the level of syntactic representations of *T* and *H*, and *iv)* the combination of different algorithms.

These experiments allowed to select the configuration achieving the highest and most stable results over training data, and train EDITS to produce the three runs described in the following section. The three submitted runs can be easily replicated by downloading the corresponding configurations from the EDITS website.

2.2 Submitted runs and results (Main task)

For the RTE6 Main task we submitted three runs.

Run 1. For the first run the system was trained on the entire training set. The learned model was obtained by using the word-overlap algorithm optimizing F-Measure of the "YES" pairs, without stop-words filtering and lexical entailment rules.

Run 2. The second run was obtained with the same configuration used for the first one, trained over the two topics for which we achieved the highest results in the training stage (namely D0914 and D0916).

Run 3. The configuration used for the third run is a slight variant of the one used for Run 2. Here, the cost of the substitution of a word in *T* with a word in *H* is the Levenshtein distance [6] between the two words (instead of 1 as in the other two runs). This solution allows for more flexible mappings between *T* and *H*, since very similar words can be substituted at a low cost even if they are not equal. Also in this case, the model was learned on topics D0914 and D0916.

The results achieved by each run, both on the training and test data, are reported in Table 1. As can be seen from the table, our best result has been achieved by Run 3 (44.71% Micro-Averaged F-measure). Even though it is based on a model learned with a very simple configuration, our best result is quite close to the best score (48.01%), and considerably above the reported median for the 48 runs submitted to the Main task (33.72%).

	Training		**Test**	
	F	Acc.	Micro-Avg. F	Macro-Avg. F
Run 1 (trained on ALL Training)	40.07	91.60	40.97	42.84
Run 2 (trained on D0914+D0916)	51.36	91.66	40.00	41.97
Run 3 (trained on D0914+D0916)	50.42	92.50	**44.71**	46.35

Table 1: Main task results

The three submissions can be easily replicated downloading the corresponding configurations from the EDITS website.

3 KBP Validation Pilot task

Given a document *D*, and a set *S* of hypotheses *S*=$\{H_1,...,H_n\}$, the KBP Validation Pilot task consists in determining if *D* entails *S*.

The task is situated in the Knowledge Base Population scenario, and aims at validating the output of the systems participating in the RTE6 KBP Slot Filling task by using Textual Entailment techniques. In this framework, *S* is a set of roughly synonymous sentences representing different linguistic realizations of a relation between a target entity, and a possible value (a.k.a. "slot-filler") of one of its attributes (a.k.a. "slots"). The assumption is that an extracted slot filler is correct if and only if the supporting document entails an hypothesis created on the basis of the slot filler.

3.1 Training the system

As in the Main task, KBP Validation training data were splitted in two portions (TRAIN and DEV) in order to perform reliable routines of training and evaluating the learned models over unseen data. The first pre-processing step consisted in performing automatic co-reference resolution on the documents using BART (the Beautiful Anaphora Resolution Toolkit [9])[2]. As a second step, we run the TextPro tagger [8] annotating words in terms of token, lemma, part-of-speech, and full morphological analysis. Then, we extracted the list of the entities in the hypothesis (e.g. *<entity>Chris Simcox<\entity>*) and we substituted in the original documents the co-reference chains (*i.e.* pronouns, anaphoric terms) solved by BART referring to those entities, with the normalized version of the entity found in the *H*s (*e.g. he/Simcox/the founder said* $\rightarrow$ *Chris Simcox said*). The resulting documents were then sentence-splitted, and used as the new training and test sets for the following steps of our experiments[3].

Finally, a large entailment corpus of *T*-*H* pairs is created, where each pre-processed sentence in the document is paired with the corresponding *H*s. To run training/evaluation routines over this huge amount of data, we experimented with the application of different types of filters trying to make the task computationally feasible with a reduction of the search space. The first filter, used for both the submitted runs, operates at the level of **documents** automatically discarding as possible entailing candidates all the sentences in a document that do not contain at least one word from the entity (*e.g.* at least "*Chris*" or "*Simcox*" for *<entity>Chris Simcox<\entity>*), and one word from the value (*e.g.* at least "*Tucson*" or "*Ariz.*" for *<value>Tucson, Ariz.<\value>*). Targeting precision, and to avoid unintended filtering of good sentences, we did not apply any further filter to the value (*e.g.* given *<entity>Chris Simcox<\entity>* and *<value>founder of the Minuteman Civil Defense Corps<\value>* we retained also sentences containing uninformative word pairs, such as *<Chris,of>*. Once this filter is applied, all the pairs for which no *T* is retained are assigned to "NO"

Then, since *H*s are automatically generated from the slot fillers, two additional filters at the level of ***T-H* pairs** have been applied to decide if a set of *H*s is *plausible*

[2] `http://bart-anaphora.org/`

[3] Due to the miscellaneous nature of the documents in the data sets, some files contain only long lists of names (people of products' names) or symbols. For these documents, namely 32 files in the training set and 316 files in the test set, no co-reference resolution was performed since BART was unable to process them.

(*i.e.* it is true in a possible world). The idea is that if this plausibility condition does not hold, the set of *H*s cannot be entailed by any document (*e.g.* the fact "*Chris Simcox is aged Paris*" is always false and cannot be supported by any document[4]). The first filter (*EVRel*, for "Entity-Values Relatedness", used for our first run) uses Latent Semantic Analysis[5] to calculate relatedness between *entities* and *values* (*e.g.* estimating the relatedness of <*entity*>*Chris Simcox*<*entity*> and <*value*>*one*<*value*> as one of the possible slot fillers for the attribute "*age*"). Once this filter is applied, all the *T-H* pairs where such relatedness is equal to 0 are assigned to "NO". The second filter (*VComp*, for "Value Compatibility", used for our second run) uses Latent Semantic Analysis to measure, for each attribute present in the dataset (*e.g.* the slot "*age*"), a relatedness threshold between all its possible values (*e.g.* the slot-fillers "*one*", "*35*", "*1972*", "*Paris*", etc.). Once this filter is applied, all the *T-H* pairs where the value is not compatible with the other values for a given attribute are assigned to "NO".

The outcomes of these experiments allowed to define the best performing configuration, and train EDITS to produce the two runs described in the following section. Both the submissions have been obtained without any modification to the system, and can be easily replicated by downloading the corresponding configurations from the EDITS website.

3.2 Submitted runs and results (KBP Validation task)

For the KBP Validation task we submitted two runs, whose results (both on training and test data) are reported in Table 2.

Run 1. For the first run the learned model was obtained by using the token edit distance algorithm optimizing F-Measure of the "YES" pairs, without stop-words filtering, using precise Wikipedia lexical entailment rules (*i.e.* with a relatedness score ≥ 0.7). As regards the aforementioned search space reduction strategies, the *"EVRel"* filter has been applied to make the task computationally feasible.

Run 2. The second run was produced with the same system configuration, but applying the *"VComp"* filter.

	Training		**Test**		
	F	Acc.	Prec	Rec	F
Run 1	21.49	62.93	20.46	33.82	**25.50**
Run 2	23.55	58.41	19.69	34.66	25.11

Table 2: KBP Validation Pilot task results

As Table 2 shows, the two runs are quite close in terms of performance, with the first run (25.5% F-measure) achieving the best score in this task. It's worth mentioning that, even with the application of filters to reduce the search space, KBP Validation proved to be a very difficult task, especially from a computational point of view (consider that the size of the tokenized test set is around 2.1Gb). In particular, parsing the test data was not feasible in reasonable time (thus making impossible experiments with some of our

[4]Under the assumption that documents describe true facts about entities.

[5]We compute LSA scores over Wikipedia in the same way described in [7].

algorithms), and the only way to produce final submissions was to split the test set in 12 portions, each of which processed on a multi-core system with the multi-threading feature provided by the latest release of EDITS.

4 Conclusion

We participated in the RTE6 Main and KBP Validation Pilot tasks with the latest release of EDITS (Edit Distance Textual Entailment Suite) the open source RTE package developed by FBK (*http://edits.fbk.eu*). In spite of the difficulty to process the two huge datasets, the efficiency and the flexibility of the system allowed for a large number of experiments to define the best system configurations for both the tasks. Unfortunately, the impossibility to parse KBP test data hindered experiments with algorithms using syntactic information. Though simple, the system configurations selected for our submissions proved to be effective, with good results in both the tasks. Our best run in the Main task achieved a Micro-Averaged F-measure of 44.71%, which is close to the reported best system (48.01%), and considerably above the reported median for all submitted runs of 33.72%. Our best run in the KBP Validation Pilot task achieved the highest score, with 25.5% F-measure.

To allow replicability of the results, and as an additional contribution to the RTE community, the configurations used for our submissions are freely downloadable from the EDITS website.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement n. 248531 (CoSyne project).

References

[1] Kouylekov, M. and Negri, M. (2010). An Open-Source Package for Recognizing Textual Entailment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010) System Demonstrations* Uppsala, Sweden, July 11-16.

[2] D. Klein and C.D. Manning. (2003) Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15* Cambridge, MA. MIT Press.

[3] Chklovski, T. and Pantel, P. (2005). VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)* Barcelona, Spain.

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

[5] Dekang Lin (1998). Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING)* Montreal, Quebec, Canada.

[6] Levenshtein, V.I. (1965). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In *Doklady Akademii Nauk* SSSR, 163(4).

[7] Mehdad, Y., Negri, M., Cabrio, E., Kouylekov, M., and Magnini, B. (2009). Using Lexical Resources in a Distance-Based Approach to RTE In *TAC 2009 Notebook Papers* Gaithersburg, MD, US.

[8] Pianta, E., Girardi, C., Zanoli, R. (2008). The TextPro tool suite. In *Proceedings of LREC, 6th Edition of the Language Resources and Evaluation Conference.* Marrakech, Morocco, 28-30 May 2008.

[9] Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A. (2008). BART: A Modular Toolkit for Coreference Resolution. In *Companion Volume of the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*. Columbus, OH. Jne 15-20.