# BnO at the NTCIR-11 English Fact Validation Task

Pascual Martínez-Gómez
Ochanomizu University,
National Institute of
Informatics, Japan
pascual@nii.ac.jp

Ran Tian
Tohoku University, Japan
tianran@ecei.tohoku.ac.jp

Yusuke Miyao
National Institute of
Informatics, Japan
yusuke@nii.ac.jp

## ABSTRACT

This paper describes the submission of BnO team to the RITE-VAL Fact Validation task [11] for English in NTCIR-11. In this submission, the BnO team formulated the fact validation as a textual entailment task, where the objective is to find a piece of text from a corpus such that it entails the stated fact. For that purpose, BnO team made use of search results retrieved by the search engine TSUBAKI as text T side of textual entailment pairs. Then, we used a logical algebraic inference system developed in [18] to test whether or not an entailment relation exists between the sentences retrieved by TSUBAKI and the hypotheses. We also tested a classifier based on Random Forests that used the output from the inference engine and other features related to TSUBAKI search results.

## Team Name

BnO

## Subtasks

Fact Validation (English)

## Keywords

textual inference, algebraic inference, statistical modeling

## 1. INTRODUCTION

In a fact validation task, the objective is to find a source of information such that it entails or contradicts the fact statement (or hypothesis). Fact validation comprises a wide set of challenges, such as assessing the accuracy or faithfulness of the information source, and assessing whether such information source logically entails the hypothesis. In this year's fact validation task, the source of information is Wikipedia, and thus we assume that the source is reliable, that there are no attempts to introduce unfaithful pieces of information, and that the source of information contains all information necessary to prove or disprove the hypothesis. With the objective of casting the formulation of the fact validation task as a textual entailment task, we also assume that the hypothesis does not entail nor contradict itself, and that other sources of information are strictly necessary to validate the hypothesis.

In principle, fact validation would require to search in a vast amount of data such as plain text or (semi-)structured databases. However, setting up a natural language interface to a database, or a search engine to retrieve relevant content from Wikipedia is not straightforward, and the task organizers provided the top 5 results of TSUBAKI [16] search engine. Each result from TSUBAKI consisted of 5 sentences, scored with a measure that quantifies the similarity between each search result and the hypothesis (the fact that needs to be validated). Thus, we also worked under the assumption that the relevant content to validate a hypothesis had to be present in the search results obtained by TSUBAKI.

From our perspective, fact validation is a very relevant end-application that requires the development and implementation of a wide array of natural language processing techniques. Specifically, it can be regarded as an exercise of integrating information extraction and textual entailment, with the purpose of enabling fact validation.

In this task, our objectives were:

1. To understand what are the main challenges of integrating information extraction and textual entailment recognition for the task of fact validation.
2. To compare our in-house logic-based textual entailment recognizer to other state-of-the-art systems.
3. To test preliminary ideas on the integration of statistical entailment recognizers with logic-based systems.

In Section 2 we briefly review other attempts for fact validation and textual entailment. In Section 3 we describe our logic-based system and our statistical classifier. Section 4 describes our experimental conditions and setup, and we present our results in Section 5. We close this paper with some conclusions and future steps in Section 6.

## 2. RELATED WORK

The task of recognizing textual entailment has attracted interest since it is a good test bed to assess extrinsically the performance and usefulness of many NLP technologies. Moreover, achieving a high performance in the recognition of textual entailment would also potentially benefit other applications of natural language processing such as text summarization, information extraction or question answering. At the NTCIR Fact Validation Task, our strategy consisted in using a textual entailment recognizer to find positive or negative entailment relations between any sentence from a corpus and the target fact (or hypothesis) that we had to validate.

Unlike other NLP applications, most successful state-of-the-art textual entailment recognizers are based on logical inference engines, being the work in [1, 2] some of the most prominent systems. These systems (and others based on

logical inference), first represent the text T and hypothesis H using a structured representation (parse trees or logic formulae), and then apply sequences of rules that aim to prove whether or not the hypothesis can be produced (or logically derived) from the text. These systems may optionally make use of external linguistic resources (such as WordNet [6]) or common background knowledge.

Many other approaches have been adopted for the task of recognizing textual inference, such as computing edit-distances between tree representations of a text T and a hypothesis H (plus a threshold) [14, 8, 12], syntactic transformations that aim to derive the syntactic structure of the hypothesis given the syntactic structure of the text [7, 15], aligning components of T and H [10, 5], and ensembles of several recognizers that attempt to take advantage of the strengths of a set of decision mechanisms [17, 19]. We refer the readers to [4] for a comprehensive description and comparison of these approaches together with an analysis of particular systems.

Our submission to NTCIR Fact Validation Task consisted in a system that performs logic inference and that makes use of external linguistic resources on-demand. Moreover, we also submitted the results of a preliminary investigation that combines the predictions of our logical system using an ensemble machine. We describe our systems in Section 3.

# 3. METHODOLOGY

Our first submission consisted in the results obtained by *tifmo*, an algebraic search engine that follows the logical inference approach. Our second submission consisted in results obtained by a statistical classifier that builds on top of the logical system, and that uses features extracted from T-H pairs together with the output from the logical system. These methods are described below.

## 3.1 Algebraic Inference

Our first submission consisted in the results obtained by *tifmo* [18], an RTE system based on the Dependency-based Compositional Semantics and logical inference. The system uses Stanford CoreNLP to parse sentences into dependency trees and resolve coreferences. Then, a rule-based conversion from the dependency trees to DCS trees is performed, to obtain a semantic representation of the sentences. Logical inferences are conveyed upon the semantic representation by *tifmo*'s algebraic inference engine, trying to deduce the semantic representation of H from the semantic representation of T.

To compensate for the strict nature of logical inference, *tifmo* also generates some on-the-fly knowledge from a T-H pair, in case H is not already proven from T. The on-the-fly knowledge is generated as DCS tree transformations from T to H, and is evaluated by some distributional similarity score, which by default uses Mikolov's word vector [13][1]. On-the-fly knowledge with a similarity score higher than a threshold (which is set to 0.4 by default) is used in logical inference, and a "Y" label is outputted if and only if H is finally proven.

To apply *tifmo* system on the fact validation task, we concatenated all sentences from search results in TSUBAKI, and use them to build a T side of T-H pairs. No extra tuning is performed on the system.

[1] `https://code.google.com/p/word2vec/`

## 3.2 Statistical Recognition

Our second submission consisted in the output produced by a statistical binary classifier. Our statistical binary classifier was a function $f : \mathcal{F}^+ \rightarrow \{\text{Yes}, \text{No}\}$ that given a list of feature values $\mathcal{F}^+$, it outputs a textual entailment judgment (Yes or No). The parameters of such function $f$ were estimated from the observation of the training and development sets. More specifically, each observation of the training or development sets consisted in a T-H pair, where H was the hypothesis (or fact) that we had to validate, and T was the concatenation of all sentences of each search result produced by TSUBAKI.

The features extracted from each T-H pair were measurements of its characteristics, such as the entailment decision produced by *tifmo*, the confidence score on such entailment decision (also produced by *tifmo*), similarity scores of each sentence in T with respect to the hypothesis (as given by TSUBAKI), and the minimum scores across each search result. TSUBAKI also produced the number of search results that a query containing the hypothesis produced, and we added this number in our feature set. The rationale was that search queries with a larger number of results could be a signal that the hypothesis (or fact) has more chances to be partially stated somewhere else. Then, we scaled the feature values of our observations in our training and development sets. A list of features used in the statistical recognition can be found in Table 1, together with a short description and the type of feature.

We used Random Forests [3] as a machine learning algorithm to estimate the parameters of the function $f$ given the feature values described above. Random Forests is a method that builds a multitude of decision trees on different partitions of the training data, and then combines the decisions into a unique decision using a majority vote. Random Forests is an ensemble method by itself, since it combines the judgments of many decision functions (decision trees). In our application, we add tifmo's judgment decision and confidence to the feature set, thus enlarging the ensemble using signals from another entailment recognizer. We selected the most relevant features using a feature importance method based on Random Forest, where the value of each feature is permuted across its domain and the impact on classification performance in out-of-bag observations is measured. Features with a positive impact (or importance) were the ones used to train the final system that was used on the test set.

# 4. EXPERIMENTAL SETUP

We used the Stanford parser to build the DCS trees, and tifmo [18] to carry out the logical inference. We set the threshold value to the linguistic similarity at 0.6, since such threshold showed good performance in past evaluations and some preliminary experiments. We used the R package `randomForest` [9] for the estimation of the statistical binary classifier and to run the feature importance, and we used 500 decision trees when training the ensemble.

In order to select the optimal number of features that lead to the highest accuracy, we run several leaving-one-out cross-validations on the training set using 10, 15, 20, 25 and 30 features. The optimal accuracy was found when using 20 features, and that number was used in testing. Surprisingly, *tifmo*'s binary entailment judgment was estimated to have a negative impact on the performance of the classifier, but

| Feature | Short description | Type |
|---|---|---|
| *tifmo*'s decision | Whether tifmo judges the T-H pair as a positive or negative entailment | Binary |
| *tifmo*'s confidence | The confidence score that tifmo produces on its own entailment decision | Real |
| $\text{Score}_{i,j}$ | Similarity score of sentence $i$ of search result $j$ w.r.t. H | Real |
| $\text{Min score}_j$ | Minimum similarity score of search result $j$ w.r.t. H | Real |
| $\text{Total search results}_j$ | Total number of search results provided by TSUBAKI for hypothesis $j$ | Integer |

**Table 1: Features used in the construction of a statistical classifier that judges whether or not any of the search results provided by TSUBAKI search engine entails the hypothesis (fact). The feature name, a short description and the type (after scaling) of the feature values are displayed.**

| system | Accuracy |
|---|---|
| prior | 62.53 |
| run 1: logical inference | 52.80 |
| run 2: statistical classifier | 61.35 |

**Table 2: Results of our systems based on logical inference and the statistical classifier. The accuracy was measured in a leaving-one-out cross-validation on the concatenation of training and development sets.**

| system | Macro $F_1$ | Accuracy |
|---|---|---|
| run 1: logical inference | 53.17 | 55.85 |
| run 2: statistical classifier | 45.29 | 52.66 |

**Table 3: Results of our systems on the test set, as provided by the task organizers. We scored first and last using run 1 and run 2, respectively.**

its confidence score proved to have a very positive impact. Other features that had a positive impact (and that were selected for the testing run) were the total number of search results, the minimum similarity scores of each search result, and the similarity scores of some sentences from the search results.

## 5. RESULTS

In our preliminary experiments, we evaluated our systems using the accuracy in a leaving-one-out cross-validation set-up on the observations from the training and the development sets. Results can be found in Table 2.

Every prediction of our systems consist in a binary {Yes, No}-judgment. The frequency of occurrence of each judgment in the training and development set is Yes at a 38% and No at a 62%. Thus, a naïve system that uses information on the prior to output always "No" would have an accuracy of 62% in the (blinded) test set assuming that the test has the same distribution on the labels Yes/No. However, we did not submit such system based on the prior as a formal run. The label distribution in the test set (released after the formal run submission) was Yes at a 39% and No at a 61%, which is very similar to the distribution of labels observed in the training and development data.

Results of our systems on the test set (as provided by the task organizers) can be found in Table 3. As we can observe, the logical inference system (run 1) achieved a higher Macro $F_1$ and accuracy with respect to our statistical classifier (run 2).

## 6. CONCLUSIONS AND FUTURE WORK

One approach to fact validation consists in searching a corpus of plain text to find potential candidate text portions (Ts) that may entail the hypothesis or fact (H). Since setting up such a search engine to retrieve candidate text portions is not a trivial task, the task organizers provided participants with 5 search results, each result consisting of 5 sentences. However, from a manual analysis on the training data, we found that, in some hypothesis (or facts), all search results did not contain enough clues for a human to judge whether there is a positive entailment relationship, without external knowledge resources.

This observation suggests that the fact validation in this shared task might be more challenging than other traditional setups of textual entailment, and that the search for possible text portions that may entail the hypothesis should not be overlooked in future applications. One of our objectives this year will focus on the development of integrated techniques that are capable of mining background knowledge on-demand, from either plain text or (semi-) structured databases, as we believe lack of knowledge and clues is one of the current bottlenecks in our systems.

As we could observe from the evaluation of the formal runs, our system based on algebraic inference performed better that our system based on the statistical classifier implemented using Random Forests. From the results, we could say that Random Forests did not manage to find a meaningful function that puts in correspondence the features values extracted from T-H pairs, and the true entailment judgment. One explanation is that the amount of data was not large enough for Random Forests to estimate appropriate parameter values; another explanation could be that the features that we extracted were not significant and did not make the contributions in the classification power that we expected. We favor the latter explanation and, in this respect, we plan to derive some more features such as the maximum similarity score (and its standard deviation) of all sentences from a single search result, and the the maximum (and standard deviation) similarity score across all sentences from all search results of a given T-H pair.

In a parallel line of research, search engines could score search results using more elaborated metrics, such as tree edit distances or partial subsumptions between logic representations of the search results and the logic representations of the hypothesis (that is used as a query). Such integration of textual entailment components into search engines would lead to obtaining search results that are semantically more related to queries, and would favor fact validation tasks. However, search engines rely on similarity scores that can be computed efficiently, which may pose a challenge.

## 7. REFERENCES


[1] R. Bar-Haim, I. Dagan, I. Greental, and E. Shnarch. Semantic inference at the lexical-syntactic level. In *Proceedings of the AAAI*, pages 871–876. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[2] J. Bos and K. Markert. Recognising textual entailment with logical inference. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 628–635. Association for Computational Linguistics, 2005.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] I. Dagan, D. Roth, M. Sammons, and F. M. Zanzotto. *Recognizing textual entailment: Models and applications*, volume 6. Morgan & Claypool Publishers, 2013.

[5] M.-C. de Marneffe, T. Grenager, B. MacCartney, D. Cer, D. Ramage, C. Kiddon, and C. D. Manning. Aligning semantic graphs for textual inference and machine reading. In *In AAAI Spring Symposium at Stanford 2007*, 2007.

[6] C. Fellbaum. *WordNet: An electronic lexical database*. Wiley Online Library, 1998.

[7] A. Fowler, B. Hauser, D. Hodges, I. Niles, A. Novischi, and J. Stephan. Applying COGEX to recognize textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 69–72, 2005.

[8] M. Kouylekov and B. Magnini. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the First Challenge Workshop Recognising Textual Entailment*, pages 17–20, 2005.

[9] A. Liaw and M. Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002.

[10] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 41–48. Association for Computational Linguistics, 2006.

[11] S. Matsuyoshi, Y. Miyao, T. Shibata, C.-J. Lin, C.-W. Shih, Y. Watanabe, and T. Mitamura. Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task. In *Proceedings of the 11th NTCIR Conference*, 2014.

[12] M. Mehdad, N. Matteo, C. Elena, K. Milen, and B. Magnini. EDITS: An open source framework for recognizing textual entailment. In *Text Analysis Conference (TAC 2009)*, 2009.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. 2013.

[14] V. Punyakanok, D. Roth, and W.-t. Yih. Natural language inference via dependency tree mapping: An application to question answering. Technical Report No. UIUCDCS-R-2004-2443, UIUC Computer Science Department, 2004.

[15] R. Raina, A. Y. Ng, and C. D. Manning. Robust textual inference via learning and abductive reasoning. In *AAAI*, pages 1099–1105, 2005.

[16] K. Shinzato, T. Shibata, D. Kawahara, and S. Kurohashi. TSUBAKI: An open search engine infrastructure for developing information access methodology. *Journal of information processing*, 20(1):216–227, 2012.

[17] M. Tatu, B. Iles, J. Slavick, A. Novischi, and D. Moldovan. Cogex at the second recognizing textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 104–109, 2006.

[18] R. Tian, Y. Miyao, and T. Matsuzaki. Logical inference on dependency-based compositional semantics. In *Proceedings of ACL*, 2014.

[19] R. Wang and G. Neumann. An divide-and-conquer strategy for recognizing textual entailment. In *Proc. of the Text Analysis Conference, Gaithersburg, MD*, 2008.