ISTI@SemEval-2 Task #8: Boosting-Based Multiway Relation Classification

Andrea Esuli, Diego Marcheggiani, Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
firstname.lastname@isti.cnr.it

Abstract

We describe a boosting-based supervised learning approach to the "Multi-Way Classification of Semantic Relations between Pairs of Nominals" task #8 of SemEval-2. Participants were asked to determine which relation, from a set of nine relations plus "Other", exists between two nominals, and also to determine the roles of the two nominals in the relation.

Our participation has focused, rather than on the choice of a rich set of features, on the classification model adopted to determine the correct assignment of relation and roles.

1 Introduction

The "Multi-Way Classification of Semantic Relations between Pairs of Nominals" (Hendrickx et al., 2010) we faced can be seen as the composition of two sub-tasks:

1. Determining which relation r, from a set of relations R (see Table 1), exists between two entities e_1 and e_2.

2. Determining the direction of the relation, i.e., determining which of $r(e_1, e_2)$ or $r(e_2, e_1)$ holds.

The set R is composed by nine "semantically determined" relations, plus a special Other relation which includes all the pairs which do not belong to any of the nine previously mentioned relations.
The two novel aspects of this task with respect to the similar task # 4 of SemEval-2007 (Girju et al., 2007) ("Classification of Semantic Relations between Nominals") are (i) the definition of the task as a "single-label" classification task and (ii) the need of determining the direction of the relation (i.e., Item 2 above).

1	Cause-Effect
2	Instrument-Agency
3	Product-Producer
4	Content-Container
5	Entity-Origin
6	Entity-Destination
7	Component-Whole
8	Member-Collection
9	Message-Topic

Table 1: The nine relations defined for the task.

The classification task described can be formalized as a *single-label* (aka "multiclass") text classification (SLTC) task, i.e., as one in which exactly one class must be picked for a given object out of a set of m available classes.

Given a set of objects D (ordered pairs of nominals, in our case) and a predefined set of *classes* (aka *labels*, or *categories*) $C = \{c_1, \ldots, c_m\}$, SLTC can be defined as the task of estimating an unknown *target function* $\Phi : D \rightarrow C$, that describes how objects ought to be classified, by means of a function $\hat{\Phi} : D \rightarrow C$ called the *classifier*[1].

In the relation classification task which is the object of this evaluation, the set C of classes is composed of 19 elements, i.e., the nine relations of Table 1, each one considered twice because it may take two possible directions, plus Other.

2 The learner

As the learner for our experiments we have used a boosting-based learner called MP-BOOST (Esuli et al., 2006). Boosting is among the classes of supervised learning devices that have obtained the best performance in several learning tasks and, at the same time, have strong justifications from computational learning theory. MP-BOOST is a

[1] Consistently with most mathematical literature we use the caret symbol (ˆ) to indicate estimation.

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 218–221,
Uppsala, Sweden, 15-16 July 2010.

variant of ADABOOST.MH (Schapire and Singer, 2000), which has been shown in (Esuli et al., 2006) to obtain considerable effectiveness improvements with respect to ADABOOST.MH.

MP-BOOST works by iteratively generating, for each class c_j, a sequence $\hat{\Phi}_1^j, \ldots, \hat{\Phi}_S^j$ of classifiers (called *weak hypotheses*). A weak hypothesis is a function $\hat{\Phi}_s^j : D \rightarrow \mathbf{R}$, where D is the set of documents and $\mathbf{R}$ is the set of real numbers. The sign of $\hat{\Phi}_s^j(d_i)$ (denoted by $sgn(\hat{\Phi}_s^j(d_i))$) represents the binary decision of $\hat{\Phi}_s^j$ on whether d_i belongs to c_j, i.e. $sgn(\hat{\Phi}_s^j(d_i)) = +1$ (resp., -1) means that d_i is believed to belong (resp., not to belong) to c_j. The absolute value of $\hat{\Phi}_s^j(d_i)$ (denoted by $|\hat{\Phi}_s^j(d_i)|$) represents instead the confidence that $\hat{\Phi}_s^j$ has in this decision, with higher values indicating higher confidence.

At each iteration s MP-BOOST tests the effectiveness of the most recently generated weak hypothesis $\hat{\Phi}_s^j$ on the training set, and uses the results to update a distribution D_s^j of weights on the training examples. The initial distribution D_1^j is uniform by default. At each iteration s all the weights $D_s^j(d_i)$ are updated, yielding $D_{s+1}^j(d_i)$, so that the weight assigned to an example correctly (resp., incorrectly) classified by $\hat{\Phi}_s^j$ is decreased (resp., increased). The weight $D_{s+1}^j(d_i)$ is thus meant to capture how ineffective $\hat{\Phi}_1^j, \ldots, \hat{\Phi}_s^j$ have been in guessing the correct c_j*-assignment* of d_i (denoted by $\Phi^j(d_i)$), i.e., in guessing whether training document d_i belongs to class c_j or not. By using this distribution, MP-BOOST generates a new weak hypothesis $\hat{\Phi}_{s+1}^j$ that concentrates on the examples with the highest weights, i.e. those that had proven harder to classify for the previous weak hypotheses.

The overall prediction on whether d_i belongs to c_j is obtained as a sum $\hat{\Phi}^j(d_i) = \sum_{s=1}^{S} \hat{\Phi}_s^j(d_i)$ of the predictions made by the weak hypotheses. The final classifier $\hat{\Phi}^j$ is thus a *committee* of S classifiers, a committee whose S members each cast a weighted vote (the vote being the binary decision $sgn(\hat{\Phi}_s^j(d_i))$, the weight being the confidence $|\hat{\Phi}_s^j(d_i)|$) on whether d_i belongs to c_j. For the final classifier $\hat{\Phi}^j$ too, $sgn(\hat{\Phi}^j(d_i))$ represents the binary decision as to whether d_i belongs to c_j, while $|\hat{\Phi}^j(d_i)|$ represents the confidence in this decision.

MP-BOOST produces a *multi-label* classifier, i.e., a classifier which independently classifies a document against each class, possibly assigning a document to multiple classes or no class at all. In order to obtain a single-label classifier, we compare the outcome of the $|C|$ binary classifiers, and the class which has obtained the highest $\hat{\Phi}^j(d_i)$ value is assigned to d_i, i.e., $\hat{\Phi}(d_i) = \arg\max_j \hat{\Phi}^j(d_i)$.

"<e1>People</e1> have been moving back into <e2>downtown</e2>."

Entity-Destination(e1,e2)

F_People FS_Peopl FH_group FP_NNP
FS1_have FS1S_have FS1H_have FS1P_VBP
FS2_been FS2S_been FS2H_be FS2P_VBN
FP3_moving FP3S_move FP3H_travel FP3P_VBG
SP3_moving SP3S_move SP3H_travel SP3P_VBG
SP2_back SP2S_back SP2H_O SP2P_RB
SP1_into SP1S_into SP1H_O SP1P_IN
S_downtown SS_downtown SH_city_district SP_NN
SS1_. SS1S_. SS1H_O SS1P_.

Table 2: A training sentence and the features extracted from it.

3 Vectorial representation

We have generated the vectorial representations of the training and test objects by extracting a number of contextual features from the text surrounding the two nominals whose relation is to be identified.

An important choice we have made is to "normalize" the representation of the two nominals with respect to the order in which they appear *in the relation*, and not in the sentence. Thus, if e_2 appears in a relation $r(e_2, e_1)$, then e_2 is considered to be the *first* (F) entity in the feature generation process and e_1 is the second (S) entity.

We have generated a number of features for each term denoting an entity and also for the three terms preceding each nominal (P1, P2, P3) and for the three terms following it (S1, S2, S3):

- T : the term itself;
- S : the stemmed version of the term, obtained using a Porter stemmer;
- P : the POS of the term, obtained using the Brill Tagger;
- H : the hypernym of the term, taken from WordNet ("O" if not available).

Features are prefixed with a proper composition of the above labels in order to identify their role in the sentence. Table 2 illustrates a sentence from the training set and its extracted features.

If an entity is composed by $k > 1$ terms, entity-specific features are generated for all the term n-grams contained in the entity, for all $n \in [1, ..., k]$. E.g., for "phone call" features are generated for the n-grams: "phone", "call", "phone_call".

In all the experiments described in this paper, MP-BOOST has been run for $S = 1000$ iterations. No feature weighting has been performed, since MP-BOOST requires binary input only.

4 Classification model

The classification model we adopted in our experiments splits the two tasks of recognizing the relation type and the one of determining the direction of the relation in two well distinct phases.

4.1 Relation type determination

Given the training set Tr of all the sentences for which the classifier outcome is known, vectorial representations (see Section 3) are built in a way that "normalizes" the direction of the relation, i.e.:

- if the training object belongs to one of the nine relevant relations, the features extracted from the documents are given proper identifiers in order to mark their role in the relation, not the order of appearance in the sentence;
- if the training object belongs to Other the *two* distinct vectorial representations are generated, one for relation Other(e_1, e_2) and one for Other(e_2, e_1).

The produced training set has thus a larger number of examples than the one actually provided. The training set provided for the task yielded 9410 training examples from the original 8000 sentences. A 10-way classifier is then trained on the vectorial representation.

4.2 Relation direction determination

The 10-way classifier is thus able to assign a relation, or the Other relation, to a sentence, but not to return the direction of the relation. The direction of the relation is determined at test time, by classifying *two* instances of each test sentence, and then combining the outcome of the two classifications in order to produce the final classification result.

More formally, given a test sentence d belonging to an unknown relation r, two vectorial representations are built: one, $d_{1,2}$, under the hypothesis that $r(e_1, e_2)$ holds, and one, $d_{2,1}$, under the hypothesis that $r(e_2, e_1)$ holds.

Both $d_{1,2}$ and $d_{2,1}$ are classified by $\hat{\Phi}$:

- if both classifications return Other, then d is assigned to Other;
- if one classification returns Other and the other returns a relation r, then r, with the proper direction determined by which vectorial representation determined the assignment, is assigned to d;
- if the two classifications return two relations $r_{1,2}$ and $r_{2,1}$ different from Other (of the same or of different relation type), then the one that obtains the highest $\hat{\Phi}$ value determines the relation and the direction to be assigned to d.

5 Experiments

We have produced two official runs.

The ISTI-2 run uses the learner, vectorial representation, and classification model described in the previous sections.

The ISTI-1 run uses the same configuration of ISTI-2, with the only difference being how the initial distribution D_1^j of the boosting method is defined. Concerning this, we followed the observations of (Schapire et al., 1998, Section 3.2) on boosting with general utility functions; the initial distribution in the ISTI-1 run is thus set to be equidistributed between the portion Tr_j^+ of positive examples of the training set and the portion Tr_j^- of negative examples, for each class j, i.e.,

$$D_1^j(d_i) = \frac{1}{2|Tr_j^+|} \quad \text{iff } d_i \in Tr_j^+ \qquad (1)$$

$$D_1^j(d_i) = \frac{1}{2|Tr_j^-|} \quad \text{iff } d_i \in Tr_j^- \qquad (2)$$

This choice of initial distribution, which gives more relevance to the less frequent type of elements of the training set (namely, the positive examples), is meant to improve the performance on highly imbalanced classes, thus improving effectiveness at the the macro-averaged level.

We have also defined a third method for an additional run, ISTI-3; unfortunately we were not able to produce it in time, and there is thus no official evaluation for this run on the test data. The method upon which the ISTI-3 run is based relies on a more "traditional" approach to the classification task, i.e., a single-label classifier trained

	Run	π^{μ}	ρ^{μ}	F_1^{μ}	π^{M}	ρ^{M}	F_1^{M}
Official results	ISTI-1	72.01%	**67.08%**	**69.46%**	71.12%	**66.24%**	**68.42%**
	ISTI-2	**73.55%**	63.54%	68.18%	**72.38%**	62.34%	66.65%
10-fold cross-validation	ISTI-1	73.60%	**69.34%**	**71.41%**	72.44%	**68.17%**	**69.95%**
	ISTI-2	**75.34%**	65.92%	70.32%	**73.96%**	64.65%	68.52%
	ISTI-3	68.52%	61.58%	64.86%	66.19%	59.75%	62.31%

Table 3: Official results (upper part), and results of the three relation classification methods when used in a 10-fold cross-validation experiment on training data (lower part). Precision, recall, and F_1 are reported as percentages for more convenience.

on the nine relations plus Other, not considering the direction, coupled with nine binary classifiers trained to determined the direction of each relation. We consider this configuration as a reasonable baseline to evaluate the impact of the original classification model adopted in the other two runs.

Table 3 summarizes the experimental results. The upper part of the table reports the official results for the two official runs. The lower part reports the results obtained by the three relation classification methods when used in a 10-fold cross-validation experiment on the training data. The evaluation measures are *precison* (π), *recall* (ρ), and the F_1 score, computed both in a $microaveraged$ ($*^{\mu}$) and a $macroaveraged$ ($*^{M}$) way (Yang, 1999).

The results for ISTI-1 and ISTI-2 in the 10-fold validation experiment are similar both in trend and in absolute value to the official results, allowing us to consider the ISTI-3 results in the 10-fold validation experiment as a good prediction of the efficacy of the ISTI-3 method on the test data. The classification model of ISTI-2, which uses an initial uniform distribution for the MP-BOOST learner as ISTI-3, improves F_1^M over ISTI-3 by 9.97%, and F_1^{μ} by 8.42%.

The use of a F_1-customized distribution in ISTI-1 results in a F_1 improvement with respect to ISTI-2 (F_1^M improves by 2.66% in official results, 2.09% in 10-fold validation results), which is mainly due to a relevant improvement in recall.

Comparing ISTI-1 with ISTI-3 the total improvement is 12.26% for F_1^M and 10.10% for F_1^{μ}.

6 Conclusion and future work

The original relation classification model we have adopted has produced a relevant improvement in efficacy with respect to a "traditional" approach.

We have not focused on the development of a rich set of features. In the future we would like to apply our classification model to the vectorial representations generated by the other participants, in order to evaluate the distinct contributions of the feature set and the classification model.

The use of a F_1-customized initial distribution for the MP-BOOST learner has also produced a relevant improvement, and it will be further investigated on more traditional text classification tasks.

References

Andrea Esuli, Tiziano Fagni, and Fabrizio Sebastiani. 2006. MP-Boost: A multiple-pivot boosting algorithm and its application to text categorization. In *Proceedings of the 13th International Symposium on String Processing and Information Retrieval (SPIRE'06)*, pages 1–12, Glasgow, UK.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 13–18, Prague, CZ. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th SIGLEX Workshop on Semantic Evaluation*, Uppsala, Sweden.

Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Robert E. Schapire, Yoram Singer, and Amit Singhal. 1998. Boosting and rocchio applied to text filtering. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 215–223, New York, NY, USA. ACM.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90.