EHU-ALM: Similarity-Feature Based Approach for Student Response Analysis

Itziar Aldabe, Montse Maritxalar
IXA NLP Group
University of Basque Country (UPV-EHU)
itziar.aldabe@ehu.es
montse.maritxalar@ehu.es

Oier Lopez de Lacalle
University of Edinburgh
IKERBASQUE,
Basque Foundation for Science
oier.lopezdelacalle@gmail.com

Abstract

We present a 5-way supervised system based on syntactic-semantic similarity features. The model deploys: Text overlap measures, WordNet-based lexical similarities, graph-based similarities, corpus-based similarities, syntactic structure overlap and predicate-argument overlap measures. These measures are applied to question, reference answer and student answer triplets. We take into account the negation in the syntactic and predicate-argument overlap measures. Our system uses the domain-specific data as one dataset to build a robust system. The results show that our system is above the median and mean on all the evaluation scenarios of the SemEval-2013 task #7.

1 Introduction

In this paper we describe our participation with a feature-based supervised system to the SemEval-2013 task #7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge (Dzikovska et al., 2013). The goal of our participation is to build a generic system that is robust enough across domains and scenarios. A domain-specific system requires new training examples when shifting to a new domain. However, domain-specific data is difficult to obtain and creating new resources is expensive.

We seek robustness by mixing the instances from BEETLE and SCIENTSBANK. We show our strategy is suitable to build a generic system that performs competitively on any domain in the 5-way task.

The paper proceeds as follows. Section 2 describes the system presenting the learning features and the runs. In Section 3 we show the optimization details, followed by the results (Section 4) and a preliminary error analysis (Section 5).

2 System description

Our system aims for robustness using the domain-specific training data as one dataset. Therefore, we do not differentiate between examples from the given domains (BEETLE and SCIENTSBANK) when training the system. In contrast, our approach dintinguishes between new questions (*unseen answer* vs. *unseen question*) as well as question types (*how*, *what* and *why*) by means of simple heuristics.

The runs are organized according to different system designs. Although all the runs use the same feature set, we split the training set to build more specialized classifiers. Training examples are grouped depending on: i) the answer is unseen; ii) the question is unseen; and iii) the question type (i.e. *what*, *how*, *why*). Each run defines a framework to explore the different ways to approach the problem. While the first run is the simplest and is the most generic in nature, the third tries to split the task into simpler problems and creates more specialized classifiers.

2.1 Similarity learning features

Our model is based on various text similarity features. Almost all of the measures are computed between question, reference answer and student answer triplets. The measures based on syntactic structure and predicate-argument overlaps are only applied to the student and reference answer pairs. In

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 580–584, Atlanta, Georgia, June 14-15, 2013. ©2013 Association for Computational Linguistics

total, we defined 30 features which can be grouped as follows:

Text overlap measures The similarity of two texts is computed based on the number of overlapping words. We obtain the similarity of two texts based on the F-Measure, the Dice Coefficient, The Cosine, and the Lesk measures. For that, we use the implementation available in the Text::Similarity package[1].

WordNet-based lexical similarities All the similarity metrics based on WordNet (Miller, 1995) follow the methodology proposed in (Mihalcea et al., 2006). For each open-class word in one of the input texts, we obtain the maximun semantic similarity or relatedness value matching the same open-class words in the other input text. The values of each matching are summed up and normalized by the length of the two input texts as explained in (Mihalcea et al., 2006). We compute the measures of Resnik, Lin, Jiang-Conrath, Leacock-Chodorow, Wu-Palmer, Banerjee-Pedersen, and Patwardhan-Pedersen provided in the WordNet::Similarity package (Patwardhan et al., 2003).

Graph-based similarities The similarity of two texts is based on a graph-based representation (Agirre and Soroa, 2009) of WordNet. The method is a two-step process: first the personalized PageRank over WordNet is computed for each text. This produces a probability distribution over WordNet. Then, the probability distributions are encoded as vectors and the cosine similarity between those vectors is calculated.

Corpus-based similarities We compute two corpus-based similarity measures: Latent Semantic Analysis (Deerwester et al., 1990) and Latent Dirichlet Allocation (Blei et al., 2003). We estimate 100 dimensions for LSA and 50 topics for LDA. Both models are obtained from a subset of the English Wikipedia following the hierarchy of science categories. We started with a small set of categories and recovered the articles below the sub-hierarchy. We only went 3 levels down to avoid noisy articles as the category system is rather flat. The similarity of two texts is the cosine similarity between the resulting vectors associated with each text in the latent space.

Syntactic structure overlap The role of syntax is studied by the use of graph subsumption based on the approach proposed in (McCarthy et al., 2008). The text is mapped into a graph with nodes representing words and links indicating syntactic dependencies between them. The similarity of two texts is computed based on the overlap of the syntactic structures. Negation is handled explicitly in the graph.

Predicate-argument overlap The similarity of two texts is computed by analyzing the overlap of the predicates and their associated semantic arguments. The system looks for verbal and nominal predicates. The similarity is also based on the approach proposed in (McCarthy et al., 2008). The graph is represented with words as nodes and the semantic role of arguments as links. First, the verbal propositions and their arguments are automatically obtained (Björkelund et al., 2009) as represented in PropBank (Palmer et al., 2005). Second, a generalization of the predicates is obtained based on VerbNet (Kipper, 2005) and NomBank (Meyers et al., 2004). Finally, the similarity of two texts is computed based on the overlap of the predicate-argument relations.

2.2 Architecture of the runs

Generic Framework RUN1 This is the simplest framework for the assessment of student answers. The system relies on a single classifier, which has been optimized on the unseen question scenario. The scenario is simulated by splitting the training set so that each question and its answers are in the same fold.

Unseen Framework RUN2 This framework relies on two classifiers. The first is tuned on an unseen answer scenario and the second is prepared for the question scenario (cf. RUN1). In order to build the unseen answer classifier, we split the training set so that answers to the same question can occur in different folders. In test time, the instance is classified depending on whether it is an unseen answer or an

[1] http://www.d.umn.edu/ tpederse/text-similarity.html

	BEETLE			SCIENTSBANK				OVERALL
	Uns-answ	Uns-qst	All	Uns-answ	Uns-qst	Uns-dom	All	All
RUN1	0.499 (6)	0.352 (7)	0.404	0.396 (7)	0.283 (4)	0.345 (3)	0.348	0.406
RUN2	0.526 (4)	0.352 (7)	0.413	0.418 (6)	0.283 (4)	0.345 (3)	0.350	0.414
RUN3	0.502 (5)	0.370 (6)	0.415	0.424 (5)	0.260 (8)	0.337 (5)	0.340	0.403
LOWEST	0.170	0.173	-	0.089	0.095	0.121	-	-
BEST	0.619	0.552	-	0.478	0.307	0.380	-	-
MEAN	0.435	0.343	-	0.341	0.240	0.267	-	-
MEDIAN	0.437	0.326	-	0.376	0.259	0.268	-	-

Table 1: 5-way results of the runs in F1 macro-average on BEETLE and SCIENTSBANK domains across different scenarios. Along with the runs, the LOWEST and the BEST system in each scenario are shown. The MEAN and MEDIAN of the dataset are also presented. Finally, the OVERALL results are showed summing up both domains. Uns-answ refers to unseen answers scenario, Uns-qst stands for unseen question, Uns-dom unseen domain and All refers to the sum of all scenarios. The run results are presented together with the ranked position in the task.

unseen question[2].

Question-type Framework RUN3 The run consists of a set of question-type expert classifiers. We divided the training set based on whether an instance reflected a *what*, *how* or *why* question. We then partitioned each question type into unseen answer and unseen question scenarios. In total, the framework deploys 6 classifiers, i.e. a test instance is classified according to the question type and scenario. We set heuristics to automatically distinguish the instance type.

3 Optimization on training set

We set a heuristic to create the training instances. For each student answer, if the matching reference answer is indicated in it, we create a triplet with the question, the student answer, and the matching reference answer. If there is no matching answer, the reference answer is randomly selected giving preference to the *best* reference answers.

Once we have a training set, we split it into different ways to simulate the scenarios described in Section 2.2. All the models are optimized using 10-fold cross-validation of the pertaining training set. For the classifiers in RUN1 and RUN2 we used 8910 training instances. For RUN3 the instances were divided as follows: 1235 instances for *how* questions, 3089 for *what* questions and 4589 for *why* questions. In total, we obtained 8 models which were distributed through the runs.

Our approach uses Support Vector Machine (Chang and Lin, 2011) to build the classifiers. As the number of features is not high, we used the gaussian kernel in order to solve the non-linear problem. The main parameters of the kernel (γ and C) were tuned using grid search over the parameter in the cross-validation setting. We focused on optimizing the F1 macro average of the classifier in order to avoid a bias towards the major classes. Each of the 8 classifiers were tuned independently.

The triplets of question, student answer and reference answer of the test instances were always created selecting the first reference answer of the given set of answers.

4 Results

A total of 8 teams participated in the 5-way task, submitting a total of 16 system runs (Dzikovska et al., 2013). Table 1 shows the performance obtained by our systems across domains and different scenarios. Our three runs ranked differently based on the evaluation scenario: beetle-uns-answ (6,4,5 rank for RUN1, RUN2, RUN3, respectively); beetle-uns-qst (7,7,6); scientsbank-uns-answ (7,6,5); scientsbank-uns-qst (4,4,8) and scientsbank-uns-dom (3,3,5). We also evaluated our runs on the entire domain (All columns) and on the whole test set (OVERALL).

The results show we built robust systems. Despite being below the best system of each evaluation scenario, the results show that the runs are competitive. All our runs are above the median and outperform the average results on each evaluation. Overall, the results attained in SCIENTSBANK are lower than in

[2] We treat unseen-domain instances as unseen-question instances.

BEETLE. This might be due to the questions and answers being longer in SCIENTSBANK, making it difficult to obtain good patterns.

As regards our runs, there is no significant overall difference. While RUN3 performs better in BEETLE unseen question and SCIENTSBANK unseen answer, in the rest of scenarios RUN2 outperforms the rest of the runs. As expected, RUN2 outperforms RUN1 in the unseen answer scenario since the former has a module specializing in unseen answers. However, although RUN3 is an ensemble of six classifiers, it is not the best run. This is probably because the training sets are not big enough.

Unseen framework (RUN2)			
	Prec	Rec	F1
correct	0.552	0.677	0.608
partially correct	0.324	0.323	0.323
contradictory	0.239	0.121	0.160
irrelevant	0.472	0.377	0.419
non domain	0.415	0.849	0.557
Macro average	0.400	0.469	0.414
Micro average	0.443	0.464	0.446

Table 2: results of the RUN2 system on a entire test set.

Table 2 shows the detailed results of the RUN2 system on the entire test set. It is noticeable the low results obtained on the *contradictory* class. This might be because the defined features are not able to model negation properly and do not deal with antonymy. Surprisingly, the *non domain* class is not the most problematic, even if the system was trained on a low number of instances.

5 Preliminary Error Analysis

We conducted a preliminary error analysis and studied some of the misclassified test instances to detect some problematic issues and to define improvements to our approach.

Example 5.1 *Sam and Jasmine were sitting on a park bench eating their lunches. A mosquito landed on Sam's arm and Sam began slapping at it. When he did that, he knocked Jasmine's soda into her lap, causing her to jump up. What was Sam's response?*

R: Sam's response was to slap the mosquito.

S1: Sam's response was to say sorry

S2: To smack the bee.

Some of the detected errors suggest that our use of syntax and lexical overlap is not sufficient to identify the correct class. Our system marks the student answer S1 from Example 5.1[3] as correct. The reference answer and the student answer share a great number of words and the dependency trees are almost identical, but not the meanings. In addition, the question contains additional information that may require other types of features to correctly classify the instance.

The predicate-argument overlap feature tries to generalize the predicate information to find similarities between verbs with the same meaning. However, our system does not always work in a correct way. The verb *smack* in the student answer S2 and the verb *slap* in the reference answer mean the same. Our system classifies the answer incorrectly. If we look at PropBank and VerbNet, we find that there is not mapping between PropBank and VerbNet for these particular verbs.

Example 5.2 *Why do you think the other terminals are being held in a different electrical state than that of the negative terminal?*

R: Terminals 4, 5 and 6 are not connected to the negative battery terminal

S1: They are connected to the positive battery terminal

We consider the negation as part of the syntactic and predicate-argument overlap measures. However, our system does not characterize the similarity between *not connected to the negative* and *connected to the positive* (Example 5.2). This type of examples suggest that the system needs to model the negation and antonyms with additional features.

In the future, further error analysis will be carried out to design features to better model the problem. We also anticipate creating a specialized feature space for each question type.

Acknowledgments

This research was partially funded by the Ber2Tek project (IE12-333), the SKaTeR project (TIN2012-38584-C06-02) and the NewsReader project (FP7-ICT-2011-8-316404).

[3] R refers to the reference answer and S1 and S2 to student answers.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL-2009)*, Athens, Greece.

Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of The Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 43–48.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May.

Scott Deerwester, Susan Dumais, Goerge Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Myroslava O. Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In **SEM 2013: The First Joint Conference on Lexical and Computational Semantics*, Atlanta, Georgia, USA, 13-14 June. Association for Computational Linguistics.

Karin Kipper. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.

Philip M. McCarthy, Vasile Rus, Scott A. Crossley, Arthur C. Graesser, and Danielle S. McNamara. 2008. Assessing forward-, reverse-, and average-entailment indices on natural language input from the intelligent tutoring system, iSTART. In D. Wilson and G. Sutcliffe, editors, *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*, pages 201–206, Menlo Park, CA: The AAAI Press.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The nombank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings the American Association for Artificial Intelligence (AAAI 2006)*, Boston.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The proposition bank: A corpus annotated with semantic role. *Computational Linguistics*, 31(1):71–106.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*.