UCD-S1: A hybrid model for detecting semantic relations between noun pairs in text

Cristina Butnariu

School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4, Ireland.
ioana.butnariu@UCD.ie

Tony Veale

School of Computer Science and Informatics
University College Dublin
Belfield, Dublin 4, Ireland.
tony.veale@UCD.ie

Abstract

We describe a supervised learning approach to categorizing inter-noun relations, based on Support Vector Machines, that builds a different classifier for each of seven semantic relations. Each model uses the same learning strategy, while a simple voting procedure based on five trained discriminators with various blends of features determines the final categorization. The features that characterize each of the noun pairs are a blend of lexical-semantic categories extracted from WordNet and several flavors of syntactic patterns extracted from various corpora, including Wikipedia and the WMTS corpus.

1 Introduction

The SemEval task for classifying inter-noun semantic relations employs seven semantic relations that are not exhaustive: Cause-Effect, Instrument-Agency, Product-Producer Origin-Entity, Purpose-Tool, Part-Whole and Content-Container. The task is to classify the relations between pairs of concepts that are part of the same syntactic structure in a given sentence. This approach employs a context-dependent classification, as opposed to usual out-of-context approaches in classifying semantic relations between noun pairs (e.g., (Turney, 2005), (Nastase *et. al*., 2006)).

Our approach is based on the Support Vector Machines learning paradigm (Vapnik, 1995), in which supervised machine learning is used to find the most salient combination of features for each semantic relation. These features include semantic generalizations of the noun-senses as encoded as WordNet (WN) hyponyms, some manually selected linguistic features (e.g., *agentive*, *gerundive*, etc.) as well as the observed relational behaviour of the given nouns in three different corpora: the collected glosses of WordNet; the collected text of Wikipedia; and the WMTS corpus.

One can find similar approaches in the literature to the semantic classification of noun compounds. Turney (2005) uses automatically extracted paraphrases to build a similarity measure between pairs of concepts, while Nastase *et. al*. (2006) proposes separate models for two different word representations when determining the semantic relation in modifier-noun compounds: a model based on the lexico-semantic aspects of words and a model that uses contextual information from corpora. Our approach is different in that we use all the available features of word representations and concept interactions in a single hybrid model.

2 System description

Our system, named the Semantic Relation Discriminator (or SRD), takes as input a set of noun pairs that are manually classified as positive/negative for a given semantic relation and produces as output a discriminator for that semantic relation. We used SRD to learn different models for each of the seven semantic relations in the classification scheme for task 4 in the SemEval Workshop. The SRD system relies on several data-resources and tools: the WordNet noun-sense hierarchy, the collected WordNet glosses for these noun-senses, the complete text of Wikipedia (downloaded June, 2005), a search engine indexing a very large corpus of text, and the WEKA Data Mining software package (version 3.5).

2.1 Feature acquisition

SRD follows four steps in acquiring features:

- *Select semantic generalizations.* For each noun-sense in a pair, SRD extracts all hypernyms at depth 8 or higher in the WordNet noun-sense hierarchy.
- *Extract syntactic phrases.* SRD looks for phrases in corpora that occur before or after each noun in a pair and which obey one of several syntactic templates. SRD also looks for joining phrases between each pair of nouns that contain 5 words or less.
- *Clean-up these phrases.* SRD lemmatizes the words in each phrase and removes function words such as articles, possessive pronouns, adjective and adverbs.
- *Record observed patterns.* For each noun pair, SRD records the following types of syntactic patterns together with their corpus frequencies: joining terms that comprise at least one verb; phrases that are composed of one verb and one preposition; and phrases that are composed of a simple verb or a phrasal verb.

2.2 Selecting the features

Due to the large number of features extracted in these steps, SRD employs five different models that use different combination of features and which pool their votes to determine a single predication for each learning task. We describe below the feature sets used for each component. The features have binary values: 1 if the feature is present for a noun pair, and 0 otherwise.

Each model employs WordNet hypernyms (from the top 8 layers of the noun hierarchy) of both noun-senses as semantic features, while models 1 and 2 employ the following additional features for each noun pair (N1, N2):

1. The most frequent syntactic patterns that appear between N1 and N2 in corpora
2. The most frequent syntactic patterns that appear between N2 and N1 in corpora

Model 1 and Model 2 differ only in the syntactic templates used to validate inter-noun patterns. Model 1 fixates on patterns that contain a verb, while Model 2 accepts patterns that contain either a preposition or a verb, or both. This yields, on average, 5,000 binary features for Model 1 for each of the seven relation types, and an average of 10,000 binary features for Model 2.

In addition to WN-derived hypernymic-features, models 3 and 4 employ the following:

1. The most frequent syntactic patterns that immediately precede N1 in a corpus
2. The most frequent syntactic patterns that immediately follow N1 in a corpus
3. The most frequent syntactic patterns that immediately precede N2 in a corpus
4. The most frequent syntactic patterns that immediately follow N2 in a corpus

In Model 3 each syntactic pattern comprises a hyphenated verb, while the syntactic patterns in Model 4 each contain a preposition or a verb. SRD generates, on average, 1,500 binary features in Model 3 and 2,500 features in Model 4 for each relation-type.

In addition to WN-derived hypernymic-features, model 5 employs the following:

1. A set of linguistic features for N1, indicating whether this noun is a nominalized verb, or whether it frequently appears in a specific semantic case role (e.g., agent).
2. The same set of linguistic features as determined for N2.

SRD generates, on average, approximately 700 binary features for each relation-type in Model 5.

2.3 Building the models

The SVM learning paradigm seems particularly suitable to our task for a number of reasons. Firstly, it behaves robustly for all seven learning tasks, ignoring the noise in the training set. This is important, since e.g., some training pairs for the Instrument-Agency relation were labeled as both true and false. Secondly, SVM has an automated mechanism for parameter tuning, which reduces the overall computational effort.

SRD employs polynomial SVMs because they appear to perform better for this task compared with simple linear SVMs or radial-basis functions.

We used the WEKA implementation of John Platt's Sequential Minimal Optimization method (Platt, 1998) to train the feature weights on all the available training data. Using SMO to train the polynomial SVM takes approx. 2.8 CPU sec. per model.

The motivation for a multiple model scheme approach comes from empirical results. SRD yields higher results relative to the five single models schemes that compose our system when evaluated using 10-fold cross validation on the training data.

3 Experiments and Results

The SemEval data-set for each of the seven semantic relations comprises 140 annotated instances for training and between 70 to 90 for testing. Each instance is manually labelled with the part of speech of each concept in a pair, as well as the WN synset-id of the intended word-sense and a sample sentential context. Neither this sample context, nor the query pattern used to originally populate the data-sets with instances, was used by SRD, so SRD's predictions fall into evaluation category B. SRD also skips those training instances where WN sense-ids are not provided, so that the actual number of training instances used ranges from 129 to 138 manually labelled examples per relation-type.

SRD's precision, recall, F-score and accuracy for each relation is given by Table 1.

	P	*R*	*F1*	*Acc*	*#t inst.*
Cause-Effect	69.8	73.2	71.4	70.0	80
Instrument-Agency	72.5	76.3	74.4	74.4	78
Product-Producer	80.6	87.1	83.7	77.4	93
Origin-Entity	60.0	50.0	54.5	63.0	81
Theme-Tool	50.0	34.5	40.8	59.2	71
Part-Whole	71.4	57.7	63.8	76.4	72
Content-Container	84.8	73.7	78.9	79.7	74
Average	**69.9**	**64.6**	**66.8**	**71.4**	**78.4**

Table1. Results for SRD across the seven learning tasks

To assess the effect of varying quantities of training data, the model was tested on different fractions of the training data: dataset B1 comprises the first quarter of the training data, dataset B2 the first half, while B3 dataset comprises the first three quarters and B4 comprises the complete training dataset. We report the behavior of SRD in predicting the unseen test data when learning from these datasets in table 2. The measures of table 2 represent an average of SRD's performance across all relation-types.

	P	*R*	*F1*	*Acc*
Dataset B1	65.4	53.3	56.4	66.2
Dataset B2	67.8	63.8	63.5	69.6
Dataset B3	71.7	64.0	66.8	71.6
Dataset B4	69.9	64.6	66.8	71.4

Table2. Results for SRD on different training datasets

3.1 Error analysis

Three types of baseline values were proposed for this task. Baseline 1 ("majority baseline") is obtained by always guessing either "true" or "false", according to whichever is the majority category in the testing data-set for the given relation. Baseline 2 ("alltrue baseline") is achieved by always guessing "true". Baseline 3 ("probmatch baseline") is obtained by randomly guessing "true" or "false" with a probability matching the distribution of "true" or "false" in the testing dataset.

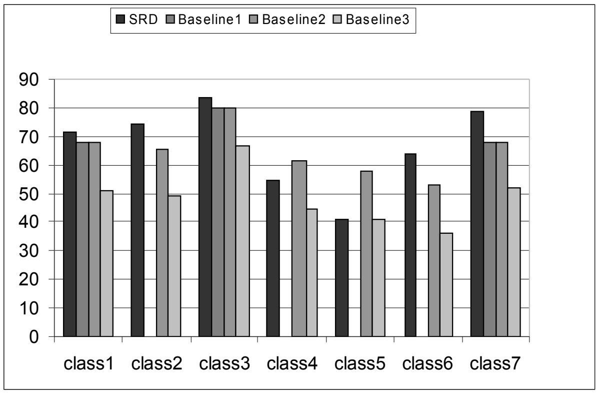

Figure1. Comparison of SRD's F-scores for each semantic relation and the corresponding baselines.

Figure 1 plots the F-scores obtained for each semantic relation. We observe that SRD has exhibits poor performance on two particular relations, Origin-Entity and Theme-Tool, denoted "class3" and "class4" in the plot of Figure 1. SRD achieves the same F-measure score as the random prediction baseline for Theme-Tool class, suggesting that the features used are simply not capable of building a discriminator for this semantic relation. SRD's F-score for Origin-Entity class is 10% higher than the random baseline, but still performs below the other two baselines. SRD's best performance is achieved for Product-Producer and Part-Whole, with an F-score 11% higher than the highest baseline.

	Feature Set1	Feature Set2	Feature Set3	Feature Set4
Cause-Effect	71.4	72.7	**75.7**	61.3
Instrument-Agency	74.4	74.6	**76.3**	72
Product-Producer	**83.7**	81.3	80.5	77
Origin-Entity	54.5	44.8	38	**61.5**
Theme-Tool	40.8	42.8	**53.8**	42.5
Part-Whole	63.8	**72.3**	62.7	60
Content-Container	**78.9**	75.6	77.1	73.2
Average	**66.8**	66.3	66.3	64

Table3. SRD F-measures using different feature sets

3.2 Improvements

One obvious problem with SRD is that we use a high-dimensional feature-space to train each model. Research in text categorization (e.g., Dumais *et al.,* 1998) shows that feature selection algorithms like information gain can identify the most productive dimensions of the feature space and simultaneously boost classification accuracy.

To explore this potential for improvement, we applied two types of feature selection filters (using WEKA): the *InfoGainAttrEval* filter that evaluates the utility of a feature by measuring information gain w.r.t. the class; and the *CfsSubsetEval* filter, which evaluates the utility of a subset of features by considering the individual predictive ability of each individually and the degree of redundancy between them collectively. Results of our experiments with SRD using different subsets of feature sets are displayed in Table 3. Set 1 is the complete set of all features. Set 2 is the subset obtained with the top n features as ranked by the *InfoGainAttrEval* filter (n is determined using 10-fold cross validation on the training data). Set 3 is a tailored feature-set created for each relation-type using the *CfsSubsetEval* filter. Set 4 is the subset of all features extracted from WN.

We find that feature-filtering boosts the performance of some learning tasks by up to 14 % (e.g., the Theme-Tool relation), but it can also decrease performance by the same amount (e.g., the Origin-Entity relation). SRD achieves its best performance -- an overall F-measure of 71.7% -- when using a feature set that is tailored to each of the semantic relation classification tasks (e.g., Set 4 (WN only) for Origin-Entity, Set 1 (all) for Product-Producer and Container-Content, Set 4 and Set 3 (relation-specific subsets) for everything else).

4 Conclusions

SRD is an SVM-based approach to classifying noun-pairs into categories that best reflect the semantic relationship underlying each pair. Without feature-filtering, SRD shows modest classification capability, performing better than the highest baselines for five of the seven relational classes. Experiments with feature filtering encourage us to try and refine SRD's feature space to focus on more discriminatory and semantically-revealing features of nouns. Feature-filtering can diminish as well as improve performance, and thus, should ideally be linked to an insightful theory of how particular features contribute to the human-understanding of noun-noun pairs. Filtering techniques provide a good basis for formulating feature-based hypotheses, but the most productive feature sets will come, we hope, from a cognitive and conceptual understanding of the processes of phrase construction, rather than from an exhaustive and largely theory-free exploration of different feature-sets.

Acknowledgments

We would like to thank Peter Turney for granting us access to the NRC copy of the WMTS.

References

Joachims, T. (1998) Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning.*

Dumais, S. T., Platt, J., Heckerman D., Sahami M., (1998) Inductive learning algorithms and representations for text categorization, *Proceedings of ACM-CIKM98*

Nastase, V., Sayyad-Shirabad, J., Sokolova, M., and Szpakowicz, S. (2006). Learning noun-modifier semantic relations with corpus-based and WordNet-based features. *In Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA.

Platt, J. (1998), Fast Training of SVMs Using Sequential Minimal Optimization, *Support Vector Machine Learning*, MIT Press, Cambridge.

Turney, P.D. (2005). Measuring semantic similarity by latent relational analysis. *In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland.

Vapnik, V. (1995). The Nature of Statistical LearningTheory, *Springer-Verlag*, New York