UTD: Classifying Semantic Relations by Combining Lexical and Semantic Resources

Bryan Rink and Sanda Harabagiu

Human Language Technology Research Institute
University of Texas at Dallas
Richardson, Texas
{bryan,sanda}@hlt.utdallas.edu

Abstract

This paper describes our system for SemEval-2010 Task 8 on multi-way classification of semantic relations between nominals. First, the type of semantic relation is classified. Then a relation type-specific classifier determines the relation direction. Classification is performed using SVM classifiers and a number of features that capture the context, semantic role affiliation, and possible pre-existing relations of the nominals. This approach achieved an F1 score of 82.19% and an accuracy of 77.92%.

1 Introduction

SemEval-2010 Task 8 evaluated the multi-way classification of semantic relations between nominals in a sentence (Hendrickx et al., 2010). Given two nominals embedded in a sentence, the task requires identifying which of the following nine semantic relations holds between the nominals: *Cause-Effect*, *Instrument-Agency*, *Product-Producer*, *Content-Container*, *Entity-Origin*, *Entity-Destination*, *Component-Whole*, *Member-Collection*, *Message-Topic*, or *Other* if no other relation is appropriate. For instance, the following sentence provides an example of the *Entity-Destination* relation:
"A small [piece]$_{E1}$ of rock landed into the [trunk]$_{E2}$."
The two nominals given for this sentence are E_1 (*piece*) and E_2 (*trunk*). This is an *Entity-Destination* relation because the piece of rock originated from outside of the trunk, but ended up there. Finally, the direction of the relation is (E_1,E_2) because E_1, the *piece*, is the *Entity* and E_2, the *trunk*, is the *Destination*.

Analysis of the training data revealed three major classes of knowledge required for recognizing semantic relations: (i) examples that require background knowledge of an existing relation between the nominals (e.g., example 5884 below), (ii) examples using background knowledge regarding the typical role of one of the nominals (e.g., example 3402), and (iii) examples that require contextual cues to disambiguate the role between the nominals (e.g., example 5710).

Example 5884 *"The Ca content in the [corn]$_{E1}$ [flour]$_{E2}$ has also a strong dependence on the pericarp thickness."*

Example 3402 *"The [rootball]$_{E1}$ was in a [crate]$_{E2}$ the size of a refrigerator, and some of the arms were over 12 feet tall."*

Example 5710 *"The seniors poured [flour]$_{E1}$ into wax [paper]$_{E2}$ and threw the items as projectiles on freshmen during a morning pep rally."*

In example 5884, the background knowledge that flour is often made or derived from corn can directly lead to the classification of the example as containing an *Entity-Origin* relation. Likewise, knowing that crates often act as containers is a strong reason for believing that example 3402 is a *Content-Container* relation. However, in example 5710, neither the combination of the nominals nor their individual affiliations lead to an obvious semantic relation. After taking the context into account, it becomes clear that this is an *Entity-Destination* relation because E_1 is going *into* E_2.

2 Approach

We cast the task of determining a semantic relation and its direction as a classification task. Rather than classifying both pieces of information (relation and direction) simultaneously, one classifier is used to determine the relation type, and then, for each relation type, a separate classifier determines the direction. We used a total of 45 feature types (henceforth: features), which

were shared among all of the direction classifiers and the one relation classifier. These feature types can be partitioned into 8 groups: lexical features, hypernyms from WordNet[1], dependency parse, PropBank parse, FrameNet parse, nominalization, predicates from TextRunner, and nominal similarity derived from the Google N-Gram data set. All features were treated as FEATURE-TYPE:VALUE pairs which were then presented to the SVM[2] classifier as a boolean feature (0 or 1).

We further group our features into the three classes described above: Contextual, Nominal affiliation, and Pre-existing relations. Table 1 illustrates sample feature values from example 117 of the training set.

3 Contextual and Lexical Features

The contextual features consist of lexical features and features based on dependency, PropBank, and FrameNet parses. For lexical features, we extract the words and parts of speech for E_1 and E_2, the words, parts of speech, and prefixes of length 5 for tokens between the nominals, and the words before and single word after E_1 and E_2 respectively. The words between the nominals can be strong indicators for the type of relation. For example the words *into*, *produced*, and *caused* are likely to occur in *Entity-Destination*, *Product-Producer*, and *Cause-Effect* relations, respectively. Using the prefixes of length 5 for the words between the nominals provides a kind of stemming (*produced* → *produ*, *caused* → *cause*).

Inspired by a feature from (Beamer et al., 2007), we extract a coarse-grained part of speech sequence for the words between the nominals. This is accomplished by building a string using the first letter of each token's Treebank POS tag. This feature is motivated by the fact that relations such as *Member-Collection* usually invoke prepositional phrases such as: *of*, *in the*, and *of various*. The corresponding POS sequences we extract are: "I", "I_D", and "I_J". Finally, we also use the number of words between the nominals as a feature because relations such as *Product-Producer* and *Entity-Origin* often have no intervening tokens (e.g., *organ builder* or *Coconut oil*).

Syntactic and semantic parses capture long distance relationships between phrases in a sentence. Instead of a traditional syntactic parser, we chose the Stanford dependency parser[3] for the simpler syntactic structure it produces. Our dependency features are based on paths in the dependency tree of length 1 and length 2. The paths encode the dependencies and words those dependencies attach to. To generalize the paths, some of the features replace verbs in the path with their top-level Levin class, as determined by running a word sense disambiguation system (Mihalcea and Csomai, 2005) followed by a lookup in VerbNet[4]. One of the features for length 2 paths generalizes further by replacing all words with their location relative to the nominals, either BEFORE, BETWEEN, or AFTER.

Consider example 117 from Table 1. The length 2 dependency path (feature *depPathLen2VerbNet*) neatly captures the fact that E_1 is the subject of a verb falling into Levin class 27, and E_2 is the direct object. Levin class 27 is the class of engender verbs, such as *cause*, *spawn*, and *generate*. This path is indicative of a *Cause-Effect* relation.

Semantic parses such as ASSERT's PropBank parse[5] and LTH's FrameNet parse[6] identify predicates in text and their semantic roles. These parses go beyond the dependency parse and identify the specific role each nominal assumes for the predicates in the sentence, so the parses should be a more reliable indicator for the relation type between nominals. We have features for the identified predicates and for the roles assigned to each nominal. Several of the features are only triggered if both nominals are arguments for the same predicate. The values from Table 1 show that the features correctly determined that E_1 and E_2 are governed by a verb of Levin class 27, and that the lexical unit is *cause.v*.

4 Nominal Role Affiliation Features

Although context can be critical to identifying the semantic relation present in some examples, in others we must bring some background knowledge to bear regarding the types of nominals involved. Knowing that a *writer* is a person provides supporting evidence for that nominal taking part in a PRODUCER role. Additionally, *writer* nominalizes the verb *write* which is classified by Levin (Levin, 1993) as an "Image creation" or "Creation and Transformation" verb. This provides further support for assigning *writer* to a PRODUCER role.

Example 117: Forward [motion]$_{E1}$ of the vehicle through the air caused a [suction]$_{E2}$ on the road draft tube.	
Feature Set	Feature Values
Lexical	e1Word=motion, e2Word=suction, e1OrE2Word={motion,suction}, wordsBetween={of, the, vehicle, through, the, air, caused, a}, posE1=NN, posE2=NN, posE1orE2=NN posBetween=I_D_N_I_D_N_V_D, distance=8, wordsOutside={Forward, on}, prefix5Between={air, cause, a, of, the, vehic, throu, the}
Dependency	depPathLen1={caused→nsubj→<E1>, caused→dobj→<E2>,...} depPathLen1VerbNet={vn:27→nsubj→<E1>, vn:27→dobj→<E2>,...} depPathLen2VerbNet={<E1>←nsubj←vn:27→dobj→<E2>}, depPathLen2Location={<E1>←nsubj←BETWEEN→dobj→<E2>}
PropBank	pbPredStem=caus, pbVerbNet=27, pbE1CoarseRole=ARG0, pbE2CoarseRole=ARG1, pbE1orE2CoarseRole={ARG1,ARG2}, pbNumPredToks=1, pbE1orE2PredHyper = {cause#v#1, create#v#1}
FrameNet	fnAnyLU={cause.v, vehicle.n, road.n}, fnAnyTarget={cause,vehicle,road}, fnE2LU=cause.v, fnE1OrE2LU=cause.v
Hypernym	hyperE1={gesture#n#2, communication#n#2, entity#n#1, ...}, hyperE2={suction#n#1, phenomenon#n#1, entity#n#1,...}, hyperE1orE2={gesture#n#2, communication#n#2, entity#n#1, suction#n#1, phenomenon#n#1, ...}, hyperBetween={quality#n#1, cause#v#1, create#v#1, ...}
NomLex-Plus	*Features did not fire*
NGrams	knnE1={motion, amendment, action, appeal, decision}, knnE2={suction, hose, pump, vacuum, nozzle}, knnE1Role=Message, knnE2Role=Component
TextRunner	trE1_E2={may result from, to contact, created, moves, applies, causes, falls below, corresponds to which}, trE2_E1={including, are moved under, will cause, according to, are effected by, repeats, can match}, trE1_E2Hyper={be#v#6, agree#v#3, cause#v#1, ensue#v#1, contact#v#1, apply#v#1, ...}

Table 1: All of the feature types and values for example 117 from the training data. Despite the errors in disambiguation the system still correctly classifies this as Cause-Effect(E$_1$,E$_2$)

We capture this background knowledge by leveraging four sources of lexical and semantic knowledge: WordNet, NomLex-Plus[7], VerbNet, and the Google N-Gram data[8].

We utilize a word sense disambiguation system (Mihalcea and Csomai, 2005) to determine the best sense for each nominal and use all of the hypernyms as a feature. Hypernyms are also determined for the words between the nominals, however only the top three levels are used as a feature. Following (Beamer et al., 2007), we also incorporate a nominalization feature for each nominal based on NomLex-Plus. Rather than use the agential information as they did, we determine the verb being nominalized and retrieve the verb's top-level Levin class from VerbNet. This reduces the sparsity problem for nominalizations while still capturing their semantics.

Our final role-affiliation features make use of the Google N-Gram data. Using the 5-grams we determined the top 1,000 words that occur most often in the context of each nominal. Nominals were then compared to each other using Jaccard similarity of their contexts and the 4 closest neighbors were retained. For each nominal, we have a feature containing the nominal itself and its 4 nearest neighbors from the training set. Additional features determine the most frequent role assigned to the neighbors. Examples of all these features can

be seen in Table 1 in the row for NGrams. The neighbors for *motion* in the table show the difficulty this feature has with ambiguity, incorrectly picking up words similar to the sense meaning a proposal for action.

5 Pre-existing Relation Features

For some examples the context and the individual nominal affiliations provide little help in determining the semantic relation, such as example 5884 from before (i.e., *corn flour*). These examples require knowledge of the interaction between the nominals and we cannot rely solely on determining the role of one nominal or the other. We turned to TextRunner (Yates et al., 2007) as a large source of background knowledge about pre-existing relations between nominals. TextRunner is a queryable database of NOUN-VERB-NOUN triples extracted from a large corpus of webpages. For example, the phrases retrieved from TextRunner for "corn ____ flour" include: "is ground into", "to make", "to obtain", and "makes". Querying in the reverse direction, for "flour ____ corn" returns phrases such as: "contain", "filled with", "comprises", and "is made from". We use the top ten phrases for the "<E$_1$> ____ <E$_2$>" query results, and also for the "<E$_2$> ____ <E$_1$>" results, forming two features. In addition, we include a feature that has all of the hypernyms for the content words in the verb phrases from the queries for the E$_1$-E$_2$ direction.

Relation	P	R	F1
Cause-Effect	89.63	89.63	89.63
Component-Whole	74.34	81.73	77.86
Content-Container	84.62	85.94	85.27
Entity-Destination	88.22	89.73	88.96
Entity-Origin	83.87	80.62	82.21
Instrument-Agency	71.83	65.38	68.46
Member-Collection	84.30	87.55	85.89
Message-Topic	81.02	85.06	82.99
Product-Producer	82.38	74.89	78.46
Other	52.97	51.10	52.02
Overall	82.25	82.28	82.19

Table 2: Overall and individual relation scores on the test set, along with precision and recall

6 Results

Our system achieved the best overall score as measured by macro-averaged F1 (for scoring details see (Hendrickx et al., 2010)) among the ten teams that participated in the semantic relation task at SemEval-2010. The results in Table 2 show the performance of the system on the test set for each relation type and the overall score.

The training data consisted of 8,000 annotated instances, including the numbered examples introduced earlier, and the test set contained 2,717 examples. To assess the learning curve for this task we trained on sets of size 1000, 2000, 4000, and 8000, obtaining test scores of 73.08, 77.02, 79.93, and 82.19, respectively. These results indicate that more training data does help, but going from 1,000 training instances to 8,000 only boosts the score by about 9 points of F-measure.

Because our approach makes use of many different features, we ran ablation tests on the 8 sets of features from Table 1 to determine which types of features contributed the most to classifying semantic relations. We evaluated all 256 (2^8) combinations of the feature sets on the training data using 10-fold cross validation. The results are shown in Table 3. The last lines of Tables 2 and 3 correspond to the system submitted for SemEval-2010 Task 8. The score on the training data is lower because the data includes examples from SemEval-2007, which has more of the harder to classify *Other* relations[9].

These tests have shown that the NomLex-Plus feature likely did not help. Further, the dependency parse feature added little beyond PropBank and FrameNet. Given the high score for the lexical feature set we split it into smaller sets to see their contributions in the top portion of Table 3. This

---

[9]To confirm this we performed a 10 fold cross validation of examples 1-7109, adding examples 7110-8000 (the 2007 data) to each training set. This resulted in an F1 of 82.18

	Feature Sets	F1
	E_1 and E_2 only	48.7
	Words between only	64.0
	E_1, E_2, and words between	72.5
	All word features (incl. before and after)	73.1
1	Lexical	73.8
2	+Hypernym	77.8
3	+FrameNet	78.9
4	+NGrams	79.7
5	-FrameNet +PropBank +TextRunner	80.5
6	+FrameNet	81.1
7	+Dependency	81.3
8	+NomLex-Plus	81.3

Table 3: Scores obtained for various sets of features on the training set. The bottom portion of the table shows the best combination containing 1 to 8 feature sets

reveals the best individual feature is for the words between the two nominals.

7 Conclusion

By combining various linguistic resources we were able to build a state of the art system for recognizing semantic relations in text. While the large training size available in SemEval-2010 Task 8 enables achieving high scores using only word-based features, richer linguistic and background-knowledge resources still provide additional aid in identifying semantic relations.

Acknowledgments

The authors would like to thank Kirk Roberts for providing code and insightful comments.

References

B. Beamer, S. Bhat, B. Chee, A. Fister, A. Rozovskaya, and R. Girju. 2007. UIUC: a knowledge-rich approach to identifying semantic relations between nominals. In *ACL SemEval07 Workshop*.

I. Hendrickx, S.N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. 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.

B. Levin. 1993. English verb classes and alternations: A preliminary investigation. *Chicago, Il*.

R. Mihalcea and A. Csomai. 2005. SenseLearner: word sense disambiguation for all words in unrestricted text. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*. ACL.

A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. 2007. TextRunner: open information extraction on the web. In *Proceedings of HLT: NAACL: Demonstrations*.