UTD-SpRL: A Joint Approach to Spatial Role Labeling

Kirk Roberts and Sanda M. Harabagiu
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083, USA
{kirk, sanda}@hlt.utdallas.edu

Abstract

We present a joint approach for recognizing spatial roles in SemEval-2012 Task 3. Candidate spatial relations, in the form of triples, are heuristically extracted from sentences with high recall. The joint classification of spatial roles is then cast as a binary classification over the candidates. This joint approach allows for a rich feature set based on the complete relation instead of individual relation arguments. Our best official submission achieves an F_1-measure of 0.573 on relation recognition, best in the task and outperforming the previous best result on the same data set (0.500).

1 Introduction

A significant amount of spatial information in natural language is encoded in spatial relationships between objects. In this paper, we present our approach for detecting the special case of spatial relations evaluated in SemEval-2012 Task 3, Spatial Role Labeling (SpRL) (Kordjamshidi et al., 2012). This task considers the most common type of spatial relationships between objects, namely those described with a spatial preposition (e.g., *in*, *on*, *over*) or a spatial phrase (e.g., *in front of*, *on the left*), referred to as the spatial INDICATOR. A spatial INDICATOR connects an object of interest (the TRAJECTOR) with a grounding location (the LANDMARK). Examples of this type of spatial relationship include:

(1) $[\text{cars}]_{\text{T}}$ parked $[\text{in front of}]_{\text{I}}$ the $[\text{house}]_{\text{L}}$.
(2) $[\text{bushes}]_{\text{T1}}$ and small $[\text{trees}]_{\text{T2}}$ $[\text{on}]_{\text{I}}$ the $[\text{hill}]_{\text{L}}$.
(3) a huge $[\text{column}]_{\text{L}}$ with a $[\text{football}]_{\text{T}}$ $[\text{on top}]_{\text{I}}$.
(4) $[\text{trees}]_{\text{T}}$ $[\text{on the right}]_{\text{I}}$. $[\emptyset]_{\text{L}}$

SpRL is a type of *semantic role labeling* (SRL) (Palmer et al., 2010), where the spatial INDICATOR is the predicate (or trigger) and the TRAJECTOR and LANDMARK are its two arguments. Previous approaches to SpRL (Kordjamshidi et al., 2011) have largely followed the commonly employed SRL pipeline: (1) find predicates (i.e., the INDICATOR), (2) recognize the predicate's syntactic constituents, and (3) classify the constituent's role (i.e., TRAJECTOR, LANDMARK, or neither). The problem with this approach is that arguments are considered largely in isolation. Consider the following:

(5) there is a picture on the wall above the bed.

This sentence contains three objects (*picture*, *wall*, and *bed*) and two INDICATORs (*on* and *above*). Since the most common spatial relation pattern is simply trajector-indicator-landmark (as in Examples (1) and (2)), the triple *wall-above-bed* is a likely candidate relation. However, the semantics of these objects invalidates the relation (i.e., walls are beside beds, ceilings are above them). Instead the correct relation is *picture-above-bed* because the preposition *above* syntactically attaches to *picture* instead of *wall*. Prepositional attachment, however, is a difficult syntactic problem solved largely through the use of semantics, so an understanding of the consistency of spatial relationships plays an important role in their recognition. Consistency checking is not possible under a pipeline approach that classifies whether *wall* as the TRAJECTOR without any knowledge of its LANDMARK.

We therefore propose an alternative to this pipeline approach that jointly decides whether a

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 419–424,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

given TRAJECTOR-INDICATOR-LANDMARK triple expresses a spatial relation. We utilize a high recall heuristic for recognizing objects capable of participating in a spatial relation as well as a lexicon of INDICATORS. All possible combinations of these arguments (including undefined LANDMARKS) are considered by a binary classifier in order to make a joint decision. This allows us to incorporate features based on all three relation elements such as the relation's semantic consistency.

2 Joint Classification

2.1 Relation Candidate Selection

Previous joint approaches to SpRL have performed poorly relative to the pipeline approach (Kordjamshidi et al., 2011). However, these approaches have issues with data imbalance: if every token could be a TRAJECTOR, LANDMARK, or INDICATOR, then even short sentences may contain thousands of negative relation candidates. Such unbalanced data sets are difficult for classifiers to reason over (Japkowicz and Stephen, 2002). To reduce this imbalance, we propose high recall heuristics to recognize candidate elements (INDICATORS, TRAJECTORS, and LANDMARKS). Since INDICATORS are taken from a closed set of prepositions and a small set of spatial phrases, we simply use a lexicon constructed from the indicators in the training data (e.g., *on*, *in front of*). Thus, our approach is not capable of detecting INDICATORS that were unseen in the training data. The effectiveness of this indicator lexicon is evaluated in Section 3.2. For TRAJECTORS and LANDMARKS, we observe that both may be considered spatial objects, which unlike INDICATORS are not a closed class of words. Instead, we consider noun phrase (NP) heads to be spatial objects. To overcome part-of-speech errors and increase recall, we incorporate three sources: (1) the NP heads from a syntactic parse tree (Klein and Manning, 2003), (2) the NP heads from a chunk parse[1], and (3) words that are marked as nouns in at least 66% of instances in Treebank (Marcus et al., 1993). This approach identifies all nouns, not just spatial nouns. But for the SemEval-2012 Task 3 data, which is composed of image descriptions, most nouns are spatial objects and no further refinements are necessary. Further heuristics (such as using WordNet (Fellbaum, 1998)) could be used to refine the set of spatial objects if other domains (such as newswire) were to be used. Our main emphasis in this step, however, is recall: by utilizing these heuristics we greatly reduce the number of negative instances while removing very few positive spatial relations. The effectiveness of our heuristics are evaluated in Section 3.2.

Once all possible spatial INDICATORS and spatial objects are marked, all possible combinations of these are formed as candidate relations. Additionally, for each spatial object and spatial INDICATOR pair, an additional candidate relation is formed with an undefined LANDMARK (such as in Example (4)).

2.2 Classification Framework

Given candidate spatial relations, we utilize a binary support vector machine (SVM) classifier to indicate which relation candidates are spatial relations. We use the LibLINEAR (Fan et al., 2008) SVM implementation, adjusting the negative outcome weight from 1.0 to 0.8 (tuned via cross-validation on the training data). This adjustment sacrifices precision for recall, but raises the overall F_1 score. For type classification (REGION, DIRECTION, and DISTANCE), we use LibLINEAR as a multi-class SVM with no weight adjustment in order to maximize accuracy. The features used in both classifiers are discussed in Sections 2.3 and 2.4.

2.3 Relation Detection Features

The difference between our two official submissions (supervised1 and supervised2) is that different sets of features were used to detect spatial relations. The features for general type classification, discussed in Section 2.4, were consistent across both submissions. Based on previous approaches to spatial role labeling, our own initial intuitions, and error analysis, we created over 100 different features, choosing the best feature set with a greedy forward/backward automated feature selection technique (Pudil et al., 1994). This greedy method iteratively chooses the best un-used feature to add to the feature set. At the end of each iteration, there is a pruning step to remove any features made redundant by the addition of the latest feature.

Before describing the individual features used in our submission, we first enumerate some basic fea-

[1] http://www.surdeanu.name/mihai/bios/

tures that form the building blocks of many of the features in our submissions (with sample feature values from Example (1)):

(BF.1) The TRAJECTOR's raw string (e.g., *cars*).
(BF.2) The LANDMARK's raw string (*house*).
(BF.3) The INDICATOR's raw string (*in_front_of*).
(BF.4) The TRAJECTOR's lemma (*car*).
(BF.5) The LANDMARK's lemma (*house*).
(BF.6) The dependency path from the TRAJECTOR to the INDICATOR (↑NSUBJ↓PREP). Uses the Stanford Dependency Parser (de Marneffe et al., 2006).
(BF.7) The dependency path from the INDICATOR to the LANDMARK (↓POBJ).

For BF.2, BF.5, and BF.7, if the relation's LANDMARK is undefined, the feature value is simply *undefined*. The features for our first submission (supervised1), in the order they were chosen by the feature selector, are as follows:

(JF1.1) The concatenation of BF.6, BF.3, and BF.7 (i.e., the dependency path from the TRAJECTOR to the LANDMARK including the INDICATOR's raw string), *for all* spatial objects related to the TRAJECTOR under consideration via a conjunction dependency relation (including the TRAJECTOR itself). For instance, TRAJECTOR_1 in Example (2) would have two feature values: ↓CONJ↓PREP↓POBJ and ↓PREP↓POBJ. Since objects connected via a conjunction should participate in the same relation, this allows the classifier to overcome the sparsity related to the low number of training instances containing a conjunction.
(JF1.2) The concatenation of BF.1, BF.3, and BF.2 (*cars::in_front_of::house*).
(JF1.3) Whether or not the LANDMARK is part of a term from the INDICATOR lexicon. Words like *front* and *side* are common LANDMARKs but may also be part of an INDICATOR as well.
(JF1.4) All the words between the left-most argument in the relation and the right-most argument (*parked*, *the*). Does not include any word in the arguments.
(JF1.5) The value of BF.7.
(JF1.6) The first word in the INDICATOR.
(JF1.7) The LANDMARK's WordNet hypernyms.
(JF1.8) The TRAJECTOR's WordNet hypernyms.
(JF1.9) Whether or not the relative order of the relation arguments in the text is INDICATOR, LANDMARK, TRAJECTOR. This order is rare and thus this feature acts as a negative indicator.
(JF1.10) Whether or not the TRAJECTOR is a prepositional object (POBJ from the dependency tree) of a preposition that is *not* the relation's INDICATOR but is in the INDICATOR lexicon. Again, this is a negative indicator.
(JF1.11) The concatenation of BF.4, BF.3, and BF.5 (*car::in_front_of::house*).
(JF1.12) The dependency path from the TRAJECTOR to the LANDMARK. Differs from JF1.1 because it does not consider conjunctions or differentiate between INDICATORs.
(JF1.13) The concatenation of BF.3 and BF.7.
(JF1.14) Whether or not the relation under consideration has an undefined LANDMARK *and* the sentence contains no spatial objects other than the TRAJECTOR under consideration. This helps to indicate relations with undefined LANDMARKs in short sentences.

The first feature selected by the automated feature selector (JF1.1) utilizes conjunctions (e.g., *and*, *or*, *either*). However, conjunctions are difficult to detect with high precision, so we decided to perform another round of feature selection without this particular feature. The chosen features were then submitted separately (supervised2):

(JF2.1) The same as JF1.2.
(JF2.2) The same as JF1.3.
(JF2.3) The same as JF1.4.
(JF2.4) The same as JF1.13.
(JF2.5) The value of BF.1.
(JF2.6) The same as JF1.5.
(JF2.7) Similar to JF1.1, but only using the concatenation of BF.6 and BF.3 (i.e., leaving out the dependency path from the INDICATOR to the LANDMARK).
(JF2.8) The same as JF1.7.
(JF2.9) The same as JF1.8.
(JF2.10) The lexical pattern from the left-most argument to the right-most argument (TRAJECTOR_*parked*_INDICATOR_*the*_LANDMARK).
(JF2.11) The raw string of the preposition in a PREP dependency relation with the TRAJECTOR *if* that preposition is not the relation's INDICATOR.
(JF2.12) The PropBank role types for each argument in the relation (TRAJECTOR=A1;INDICATOR=AM_LOC;LANDMARK=AM_LOC). Uses SENNA (Collobert and Weston, 2009) for the PropBank parse.
(JF2.13) The same as JF1.14.
(JF2.14) The concatenation of BF.4, BF.3, and BF.5.
(JF2.15) The same as JF1.10, but with no requirement to be in the INDICATOR lexicon.

2.4 Type Classification Features

After joint detection of a relation's arguments, a separate classifier determines the relation's general type. The features used to classify a relation's general type (REGION, DIRECTION, and DISTANCE) were also selected using an automated feature selector from the same set of features. Both submissions (supervised1 and supervised2) utilized these

	supervised1			supervised2		
Label	Precision	Recall	F_1	Precision	Recall	F_1
TRAJECTOR	0.731	0.621	0.672	0.782	0.646	0.707
LANDMARK	0.871	0.645	0.741	0.894	0.680	0.772
INDICATOR	0.928	0.712	0.806	0.940	0.732	0.823
Relation	0.567	0.500	0.531	0.610	0.540	0.573
Relation + Type	0.561	0.494	0.526	0.603	0.534	0.566

Table 1: Official results for submissions.

features. The following features were used for classifying a spatial relation's general type:

(TF.1) The last word of the INDICATOR.
(TF.2) The value of BF.3.
(TF.3) The value of BF.5.
(TF.4) The same as JF1.3.
(TF.5) The same as JF2.10.

3 Evaluation

3.1 Official Submission

The official results for both of our submissions is shown in Table 1. The argument-specific results for TRAJECTORS, LANDMARKS, and INDICATORS are difficult to interpret in the joint approach. In a pipeline method, these usually indicate the performance of individual classifiers, but in our approach these results are simply a derivative of our joint classification output. The first submission (supervised1) achieved a triple F_1 of 0.531 for relation detection and 0.526 when the general type is included. Our second submission (supervised2) performed better, with an F_1 of 0.573 for relation detection and 0.566 when the general type is included. This suggests that the feature JF1.1, even though it is the best individual feature, introduces a significant amount of noise.

The only result to compare our official submissions to is that of Kordjamshidi et al. (2011), who utilize a pipeline approach. Their method has a relation detection F_1 of 0.500 (they do not report a score with general type). We further compare our method with theirs in Section 4.

3.2 Relation Candidate Evaluation

The heuristics described in Section 2.1 that enable joint classification were tuned for the training data, but their recall on the test data places a strict upper bound on the recall to our overall approach. It is therefore important to understand the performance loss that occurs at this step.

Table 2 shows the performance of our heuristics on the training and test data. The spatial INDICATOR lexicon has perfect recall on the training data because it was built from this data set. However, it performs at only 0.951 recall on the test data, as almost 5% of the INDICATORS in the test data were not seen in the training data. Most of these are phrasal verbs (e.g., *sailing over*) or include the modifier *very* (e.g., *to the very left*). Our spatial object recognizer performed better, only dropping from 0.998 (2 errors) to 0.989 (16 errors). Some of these errors resulted from mis-spellings (e.g., *housed* instead of *houses*), non-head spatial objects (*mountain* from the NP *mountain landscape*), NPs containing conjunctions (*trees* in *two palm trees, lamps and flags*, which gets marked as one simple NP), as well as parser errors. The significant drop in precision for both spatial indicators and objects is an additional concern. This does not indicate the extracted items were not valid as potential indicators or objects, but rather that no gold relation contained them. As explained in Section 4, this is likely caused by the disparity in sentence length: longer sentences result in more matches, but not necessarily more relations. As evidence of this, despite the training and test data containing almost the same number of sentences, there are 36% more spatial indicators and 20% more spatial objects in the test set.

3.3 Further Experiments

After the evaluation deadline, the task organizers provided the gold test data, allowing us to perform additional experiments. In this process we found several annotation errors which we needed to fix in order to process our gold results. These errors were largely annotations that were given an incorrect token index, resulting in the annotation text not matching the referenced text. These fixes increased our performance, shown on Table 3, improving relation detection for the supervised2 feature set from 0.573

		#	Precision	Recall	F_1
Spatial Indicators	Train	1,488	0.448	1.000	0.619
	Test	2,335	0.328	0.951	0.487
Spatial Objects	Train	2,974	0.448	0.998	0.618
	Test	3,704	0.387	0.989	0.556

Table 2: Results of relation candidate selection heuristics.

Data	Precision	Recall	F_1
Train/Test	0.644	0.556	0.597
Train/Test -NSI	0.644	0.582	0.611
Train CV	0.824	0.743	0.781
Test CV	0.745	0.639	0.688
Train+Test CV	0.774	0.680	0.724

Table 3: Additional experiments on corrected test data using the supervised2 data set. -NSI indicates that the gold spatial INDICATORs that are not in the lexicon are removed. CV indicates 10-fold cross validation.

to 0.597. We use this updated data set for the following experiments. While the results aren't comparable to other methods, the goal of these experiments is to analyze our system under various configurations by their relative performance.

Table 3 also shows a 10-fold cross validation performance on 3 data sets: (1) the training data, (2) the test data, and (3) both the training and test data. While our feature set is tuned to the training data, the test data is clearly more difficult. Section 4 discusses the differences between the training and test data that may lead to such a performance reduction.

Since our lexicon of spatial INDICATORs was built from the training data, our method will not recognize any relations that use unseen INDICATORs. To differentiate between how our method performs on the full test data and just those INDICATORs that are in the lexicon, we removed the 39 gold relations with unseen INDICATORs and re-tested the system. As can be seen in Table 3 (under -NSI), this improves recall by 2.6 points.

3.4 Feature Experiments

To estimate the contribution of our features, we performed an additive experiment to see how each feature contributes to the overall test score. Table 4 shows the feature contributions based on the order they were added by the feature selector. For many of the features the score goes down when added. However, without these features, the final score would drop to 0.578, indicating they still provide valuable information in the context of the other features. Table 5 shows performance on the updated test set

Feature	Precision	Recall	F_1
JF2.1	0.333	0.156	**0.212**
+JF2.2	0.347	0.126	0.185
+JF2.3	0.708	0.115	**0.197**
+JF2.4	0.555	0.294	**0.384**
+JF2.5	0.636	0.402	**0.493**
+JF2.6	0.590	0.414	0.486
+JF2.7	0.621	0.553	**0.585**
+JF2.8	0.614	0.568	**0.590**
+JF2.9	0.573	0.568	0.571
+JF2.10	0.612	0.547	**0.578**
+JF2.11	0.625	0.571	**0.597**
+JF2.12	0.660	0.536	0.592
+JF2.13	0.633	0.573	**0.601**
+JF2.14	0.642	0.563	0.600
+JF2.15	0.644	0.556	0.597

Table 4: Additive feature experiment results using the supervised2 features. Bold indicates increases in F_1 over the previous feature set.

Feature	Precision	Recall	F_1
$\emptyset$	0.644	0.556	0.597
JF2.1	0.627	0.571	**0.598**
JF2.2	0.629	0.542	0.582
JF2.3	0.540	0.494	0.516
JF2.4	0.591	0.412	0.485
JF2.5	0.631	0.558	0.592
JF2.6	0.657	0.515	0.577
JF2.7	0.636	0.547	0.589
JF2.8	0.641	0.562	**0.599**
JF2.9	0.678	0.539	**0.601**
JF2.10	0.607	0.569	0.587
JF2.11	0.640	0.565	**0.600**
JF2.12	0.646	0.566	**0.603**
JF2.13	0.646	0.553	0.596
JF2.14	0.618	0.572	0.594
JF2.15	0.642	0.563	**0.600**

Table 5: Results when individual features from the supervised2 submission are removed. Bold indicates improvement when the feature is removed.

when individual features are removed. Here, six features that were useful on the training data did not prove useful on the test data.

4 Discussion

The only available work against which our method may be compared is that of Kordjamshidi et al. (2011). They propose both a pipeline and joint approach to SpRL. In their case, their pipeline approach performs better than their joint approach. Joint approaches increase data sparsity, so their greatest value is in the ability to use a richer set of features that describe the relationships between the arguments. Kordjamshidi et al. (2011) furthermore

did not employ heuristics to select relation candidates such as those in Section 2.1. Given this difference it is difficult to assert that a joint approach is better with complete certainty, but we believe the ability to analyze the consistency of the entire relation provides a significant advantage. Many of our features (JF2.1, JF2.3, JF2.10, JF2.12, JF2.13, and JF2.14) were of this joint type.

The drop in performance from the training data to the test data is significant. The possibility that this is entirely due to over-training is dispelled by the cross validation results in Table 3. While different features might work better on the test set, they are unlikely to overcome the cross validation difference of 9.3 points (0.781 vs. 0.688). Much of this comes from the recall limit due to the use of the spatial indicator lexicon. The other significant cause of performance degradation seems to be caused by sentence length and complexity. The test sentences are longer (18 tokens vs. 15 tokens in the training data), and have far more conjunctions (389 *and* tokens vs. 256), indicating greater syntactic complexity. But the largest difference is the number of relation candidates generated by the heuristics: 60,377 relation candidates from the training data vs. 167,925 relation candidates from the test data (the data sets are roughly the same size: 600 training and 613 test sentences). The drop of precision in spatial objects in Table 2 reflects this as well. Since the number of candidate relations is quadratic in the number of spatial objects, it is likely that just a few, long sentences result in this dramatic increase in the number of candidates.

Since more general domains (such as newswire) are likely to have this problem as well, one important area of future work is the reduction of the number of relation candidates (increasing precision) while still maintaining near-perfect recall.

5 Conclusion

We have presented a joint approach for recognizing spatial roles in SemEval-2012 Task 3. Our approach improves over previous attempts at joint classification by extracting a more precise (but still extremely high recall) set of relation candidates, allowing binary classification on a more balanced data set. This joint approach allowed for a rich set of features based on all the relation's arguments. Our best official submission achieved an F_1-measure of 0.573 on relation recognition, best in the task and outperforming all previous work.

Acknowledgments

The authors would like to thank the SemEval-2012 Task 3 organizers for their work preparing the data set and organizing the task.

References

Ronan Collobert and Jason Weston. 2009. Deep Learning in Natural Language Processing. Tutorial at NIPS.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5).

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.

Parisa Kordjamshidi, Martijn Van Otterlo, and Marie-Francine Moens. 2011. Spatial Role Labeling: Towards Extraction of Spatial Relations from Natural Language. *ACM Transactions on Speech and Language Processing*, 8(3).

Parisa Kordjamshidi, Steven Bethard, and Marie-Francine Moens. 2012. SemEval-2012 Task 3: Spatial Role Labeling. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval)*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic Role Labeling*. Morgan and Claypool.

Pavel Pudil, Jana Novovičová, and Josef Kittler. 1994. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125.