Automatic Frame Extraction from Sentences

Martin Scaiano and Diana Inkpen

School of Information Technology and Engineering,
University of Ottawa
{mscai056@uottawa.ca, diana@site.uottawa.ca}

Abstract. We present a method for automatic extraction of frames from .a dependency graph. Our method uses machine learning applied to a dependency tree to assign frames and assign frame elements. The system is evaluated by cross-validation on FrameNet sentences, and also on the test data from the SemEval 2007 task 19. Our system is intended for use in natural language processing applications such as summarization, entailment, and novelty detection.

1 Introduction

Many natural language processing tasks could benefit from new algorithms which use enhanced semantic structures and relations. Researchers have been successfully applying semantic role labeling (SRL) to tasks such as question answering [1], machine translation [2], and summarization [3]. The next step is a deeper semantic representation; but before such as representation can be used, a method for automatically creating this representation is needed. The goal of this paper is to describe a new method for this task.

Our semantic representation is based on frames [4] which represent events, objects, and situations. The specific variations and roles of a frame are defined as frame elements. This has the benefit of representing a particular situation independent of how it is expressed or phrased. Consider purchasing something; some common parameters (frame elements) are the buyer, the seller, the object in question (theme), and the price. The following three sentences represent the same situation but use different words and structures:

1. Bob bought a red car from Mary.
2. Mary’s red car was sold to Bob.
3. Bob purchased a car from Mary, that was red.

Our system relies on Berkeley FrameNet [5] for frame definitions. FrameNet is a resource that provides definitions of about 800 frames, spanning about 10000 lexical units (words that evoke a particular frame). FrameNet also provides annotated examples of the frames and lexical units being used in English. FrameNet is currently the most comprehensive resource for frame definitions and one of the few resources for semantic role labeling.

Figure 1[1] shows the result of our system for the first of the above sentences (the other sentences would have nearly identical representations)[2]. The rectangles represent frames from FrameNet, the octagons represent people, while the ovals represent words which are functioning as values for frame elements. Each edge (arrow) represents a frame element and it is labeled. The text in the top of each node represents the type (frame name, person, or value) while the text below indicates the word which the node was derived from.

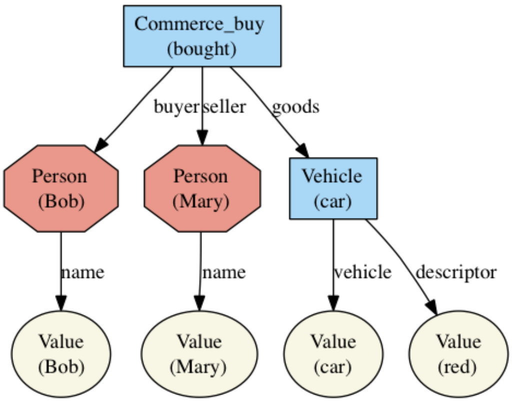

Figure 1. Frame representation of the sentence “Bob bought a red car from Mary.”

1.1 Semantic Role Labeling

Semantic role labeling is the operation of assigning the semantic relation between words. Usually roles are labeled for predicates of verbs; some general-purpose roles are as follows: agent, patient, instrument, time, location, and frequency.

The set of semantic roles can differ from resource to resource. FrameNet uses role labels (called frame elements) that are specific to the frame being annotated, while resources like PropBank [6] and VerbNet [7] provide only a small set of general-purpose roles.

The first statistical SRL system was developed by Gildea and Jurafsky [8] using FrameNet; subsequent research lead to the development of VerbNet and PropBank.

1.2 SemEval 2007 Task 19: Frame Semantic Structure Extraction

At SemEval 2007, task 19 [9] was dedicated to frame extraction. The task relied on FrameNet but provided additional training data for systems. The competition involved assigning in each sentence, which frames from the training data and FrameNet were evoked by the given lexical units, and subsequently assigning frame elements.

[1] We used Graphviz to visualize the xml output of our system. http://www.graphviz.org/

[2] FrameNet represents buying and selling as different frames though they inherit from common super frame

The resulting labeled texts from each system were evaluated based on precision and recall of detecting all the expected frames in each of the three texts. Our system was not implemented at the time of the competition, but it can be compared, for evaluation purposes, to the two participating systems, using the same data.

The UTD system [10] applies Support Vector Machines (SVM) and Maximum Entropy (ME) to the tasks of frame disambiguation and frame element assignment. Their system uses established methods for disambiguation [11] and when an inadequate amount training data is available (less than 5 sentences) the system randomly chooses the sense. The system uses established features and methods for identifying grammatical functions, frame element boundaries, and assigning frame elements.

The LTH system [12] uses a dependency parser for its initial representation. Before doing frame disambiguation, the system applies a small set of rules to filter out words which tend to cause performance degradation (in particular prepositions). Frame disambiguation is done using a SVM classifier on the resulting words. The authors extended the collection of lexical units for various frames by using WordNet [13]. Assigning frame element is done in the dependency tree using features from the target word, and some immediate related words (parent and children of the node).

The CLR system participated only in the assignment frame task. Its method of assigning frames was based on heuristic rules [14].

Our system is very different from UTD and CLR, and it is similar to LTH in some aspects, including the fact that our initial data structure is a dependency parse, except that our dependency trees are augmented with shallow semantic information. Unique to our system is the idea of using the completely annotated texts from SemEval 2007 and FrameNet to train classifiers with negative frame assignment (identifying when a word does not evoke a frame) instead of using special rules to remove unlikely frames or words. Our method's frame assignment features differ from the LTH and UTD features, in particular the use of a boolean list of evokable[3] frames and the boolean list of child nodes. Also, we tested several machine-learning models for assigning frame element (one model per lexical unit, one model per frame, and one model per location per frame) which are described below.

2 Method Description

Like most SRL methods, our method relies on machine learning for most of the decisions[4]. Our method accomplishes the following sub-tasks, each of which is described in more detail in the following subsections:

1. Initial sentence parsing;
2. Frame assignment;
3. Identification of people, entities, locations and other special types;
4. Role recognition (assignment of frame elements).

[3] Frames, which may possibly be evoked, based on matching lexical units from a given text.

[4] Our system is implemented in Java so that it can easily interfaced with Weka [15]. FrameNet provides no particular API, only xml files containing the data, so we implemented our own Java interface.

2.1 Initial Sentence Parsing

Typically SRL systems use a syntactic parser for the initial parsing; our method's initial parse is obtained with the Machinese Semantic parser by Connexor Oy[5] [16], which produces a dependency tree augmented with identification of named entities and their classes, and shallow semantic types for a variety of nouns.

The semantic parser provides about 30 dependency relations, which form a good foundation for assigning frame elements. Additionally, the parser does an effective job of identifying named entities, and their semantic classes (location, person, country, etc.).

2.2 Assigning of Frames

Our frame assignment method uses machine learning classifiers from Weka (Waikato Environment for Knowledge Analysis) [15] applied to features extracted from the parse tree. We tried Naïve Bayes (NB), Decision Trees (J48) and SVM (SMO) from Weka, using default parameter settings, and found all results to be very close for this task, as shown in Table 1. We chose these three classifiers in order to test a classifier that is known to work well with text (NB), a classifier whose output is understandable to human readers (J48), and one that is known to produce good classification results for many tasks (SVM).

The semantic parser lemmatizes each word, then a machine learning model specific to that lemma is used to classify what frame, if any, the word evokes.

The training data for the models is extracted from FrameNet's annotated examples and from SemEval 2007 task 19. Because the SemEval 2007 training was completely annotated, including words that were not frame evoking words or frame elements, we used it to extract negative examples (examples of lexical units that did not evoke any frames) for training.

Models were produced for each lemmatized head word of all lexical units. Each model determined if particular word should evoke a frame and if so which frame.

Some of the lexical units in FrameNet have very few or no training examples, thus we choose to train the frame assignment models on all examples from all lexical units which could evoke the given frame. This approach showed no decline in precision and recall, and allows the system to operate on lexical units which would not have had enough training data.

The features used for the frame assignment are as follows:

1. A boolean list for each possibly relevant frame for this lemma. The value defaults to false, unless the lexical unit in its entirety (all lexemes) for a given frame matches in the given example. Models are selected by simply matching a single word (the head word) though any lexical units containing the given head word may not fully match, thus this feature should assist the classifier in filtering inappropriate lexical units and thus inappropriate frames. This feature increased both precision and recall.

[5] http://www.connexor.eu

2. The dependency relation (as assigned by Machinese Semantics) of the word in question. This feature alone provides good results for assigning frames.
3. A boolean list of all child dependency relations, with the value being true if that dependency relation is present in the given sentence, otherwise false. This feature is motivated by the idea that neighboring words could help disambiguate frames. This feature provided a 10% increase in recall.

Table 1. 10-fold cross-validation results for frame assignment. A baseline classifier that always chooses the most frequent frame produces results of 74%; therefore our results are a significant improvement.

	Precision	Recall	F-Measure
Naïve Bayes	96%	99%	97.5%
Decision Trees	96%	99%	97.5%
SVM	97%	99%	98%

2.3 Identification of Entities

Although there are various techniques and systems available for identifying named entities, our system relies on the semantic parser's ability to identify entities. When a word is tagged as an entity, special rules convert the words into a final representation. The rules are as follows:

- If a word is tagged as a named entity and specifically a location, then a special location object is made; the tagging will also indicates the type of location, i.e., city, country, continent.
- If a word is tagged as a named entity and specifically a person, male or female, then a special person object is created. The word that was tagged as named entity is labeled as a "name" associated with the person. Any immediately adjacent words which are also named entities are added to the person frame as "name" labels. Future work is needed to determine how to combine our representation of people and entities with those from FrameNet.
- If a word is tagged as named entity but is not of a human type, we create a special general-purpose entity object. This object may represent an organization, person, location or other named entity. Like with the person object, a "name" is associated with the object and any adjacent words that are named entities.
- If a word is tagged as a human (but not a named entity) we still create a person object for the word. This often occurs with words like, "he", "she", and "you". The parser can recognize a variety of words as being people without them being a named entity. This does not provide any special benefits during evaluation, though it is our belief that this will keep the representation uniform and be more useful during future processing.

2.4 Assigning Frame Elements

Assigning frame elements is effectively the SRL task, a topic that has been extensively researched in recent years.

Our system assigns frame elements using machine learning classifiers trained on the FrameNet annotated corpus. Gildea and Jurafsky were the first to implement semantic role labeling for FrameNet [8], and the task has since been studied frequently. Our system assigns frame elements from a dependency tree provided by the Machinese Semantic parser. Our method is significantly distinct from other researchers because our system starts with a dependency tree, which provides guidance for semantic role labeling.

To assign frame elements for a particular frame the system applies a classifier to each of the words surrounding the frame evoking word. Each word is classified with either a frame element or "none", indicating no frame element and no association. 50% of all frame elements are immediate children of the frame evoking word in the dependency tree. Of all child relations about 50% are themselves frame elements.

Our method considers any word within two edges in dependency graph; thus grandparents, parents, siblings, children and grandchildren are all considered possible frame element.

From the FrameNet annotated data and the SemEval 2007 task 19 data, 978,180 training examples were extracted; among them, 225,529 of the examples were positive (of actual frame elements), while the remainder were negative examples (not frame elements). Table 2 shows the distribution of frame elements and negative examples by location in the dependency tree relative to a frame word.

About 1000 examples of frame elements could not be found within 2 edges of the frame. All the missing frame elements were in incomplete parse trees (cases when the parser had problems parsing the sentences, which resulted in a fragmented parse tree). We believe that, had the parse been complete, then most frame elements in those examples would have been found within two edges of a frame word.

Table 2. Distribution of frame elements by location relative to the frame word. Positive class are actually frame elements while negative class are not frame elements.

	Grandparents	Parents	Siblings	Children	Grandchildren	Total
Positive	1,763	16,323	50,484	146,405	10,555	225,529
Negative	49,633	77,863	224,606	148,929	251,620	752,651

The classifier is configured so that one word may only act as a single frame element, though on rare occasions a single word may actually represent two frame elements.

A frame evoking word may also be a frame element of it's own frame. This is common of general-purpose frames, such as food, where the word may specify a specific sub-type of frame, such as spaghetti, or sandwich.

When training our classifier, we experimented with several machine-learning configurations, but our best results were found by training one model per frame, which is then applied to each word being classified. The other configurations that we tried are one model per lexical unit and one model per location per frame. One model per lexical unit was meant to address the possible reversal of subject and object in some frames, but may have suffered from less training data than one model per frame. One model per location per frame involved sub-dividing the task of assigning frame elements by their location relative to the frame evoking word in the dependency tree: grandparent, parent, sibling, child, grandchild. It was thought that locations carried consistent classification properties, but this configuration performed worse than one model per frame, which is more general.

During development, we tested several features including the lexical unit that evoked the frame, part of speech, voice of the verb, location of the word in question (relative to frame word in the parse tree), dependency relation of the frame evoking word and frame element word, leading prepositions, semantic classes of words, and if the frame element word is an animate object.

The features that provided the best balance between precision and recall were the dependency relation of the frame evoking word, the dependency relation of the classified word and the dependency relation of the intermediate word, if one existed. Intermediate words exist for locations such as grandchildren which have an intermediate word that is a child of the frame evoking word; grandparent which has an intermediate word that is the parent word; and siblings words which have an intermediate word that is the parent word.

We were surprised to find that using different classifiers did not provide large differences in our results. There are small trade-offs between precision and recall, as shown in Table 3. Perhaps small improvements can be obtained in future work by combining them in an ensemble, but our focus was of finding the best features and generic models for classification.

Table 3. 10-fold cross validation results for frame element classification.

	Precision	Recall	F-measure
Naïve Bayes	74%	52%	61.0%
Decision Trees	82%	45%	58.1%
SVM	82%	43%	56.8%

3 Evaluation on Additional Test Data

The cross-validated results are very good, as is often the case when the validation data is similar in nature to the training data. Therefore, we decided that further evaluation was needed. The SemEval 2007 task 19 was a frame extraction task and could provide data for evaluation. Plain text was provided to the competing systems; each system was required to assign all frames, tag various entities, and assign frame elements in a specific output format.

Three systems entered the task for the frame assignment and entity-tagging component. Only two of the systems completed the entire task of assigning frame elements.

3.1 Evaluation Issues

Our system was not directly intended to support the required output for the SemEval 2007 task 19 and thus some adjustments to our system and to the evaluation script were required.

While developing our system we found that the FrameNet annotated data contained various inconsistencies in the frame elements names (capitalization, use of space and underscores, and even name mismatches). For consistency our system adjusts all frame element names to use spaces and lower case letters. The evaluation script used a case sensitive comparison of frame element names. Since the case of our frame elements did not necessarily match that of the gold standard, we matched all the frame elements in lower case.

The evaluation of results was founded on the assumption that tokenization would be consistent between the gold standard and the system. Our system's tokenization was significantly different from the expected tokenization and a method for conversion between tokens was required. This conversion is not exact, and occasionally leads to correct frames and frame elements being considered incorrect.

Unlike FrameNet and the expected output of SemEval 2007 task 19, our system does not select frame elements as sequences of text but selects an object (frame, entity or word representing a value) from the final representation as frame elements; this is best show in Figure 1. When the results of our system are evaluated on the task data, some of the correct frame elements are considered incorrect for having differing boundaries.

The training texts contained new frames and frame elements that were not available in FrameNet. This was intended to test a systems ability to include new frames from annotated examples. Our system easily added the new frames from the training data into the frame assignment models, but the frame element assignment models were not updated to work for these new frames, therefore our system could not get correct labels in these cases.

All the difficulties above are estimated to cause no more than 3% discrepancy in precision and recall.

3.2 Results

The SemEval task was evaluated on three texts, each having a different topic and different frequencies of frames. The task proved to be much more difficult than the task of extracting frames from FrameNet examples, which contain annotations only for one frame for each example.

Our final results for frame assignment compared to the other systems are shown in Table 4. Our system's recall and F-measure are significantly lower than that of the other two systems, but our precision is comparable.

The combined results of frame and frame element assignment are shown in Table 5. Our system's recall and F-measure still tend to be the weakest, but our precision is the best in two of the three evaluations.

This evaluation has shown that our system is a good foundation for our future work. Since our final goal is not simply to extract frames, but to apply the representation to other tasks, we can now focus on optimizing and improving our system for those tasks.

4 Conclusions and Future Work

We have developed a method for frame element extraction that has good precision and should be usable for future research. We plan many improvements in future work.

Cross-validation was an inadequate quality measure for the machine learning models. For future evaluations we are considering using more held-out texts, such as the 3 texts from SemEval 2007 and any completely-annotated texts from FrameNet. Using hold-out texts seems to provide a better evaluation of how the system will work on new domains and texts with varying distributional patterns of frames and frame elements.

The use of dependency relations as foundation for SRL (assigning frame elements in our system) has produced excellent precision. Although numerous feature combinations were tested the most important features were the dependency relations provided by the initial parser, when used in conjunction with the features of the neighboring words.

Future work on the system includes evaluating new features for assigning frame elements, such as a frame evoked by a frame element word (if one exists) and considerations for any intermediate frame elements. Also semantic type checking using WordNet is a possibility.

Extending FrameNet with information from other sources, to increase its coverage [17], is another direction for future work. We are particularly concerned with adding more training data for the frames that have too few manually annotated examples.

We plan to focus on how to use the extracted frame-based representations in tasks such as summarization, novelty detection, and entailment, exploiting the fact that our representations have high precision. When recall is insufficient, we can fall back of the shallow methods based on syntactic dependencies that we used before for the three mentioned tasks.

Table 4. Frame assignment results on the SemEval 2007 Task 19 test data, which consisted in three texts, entitled Dublin, China and Work. The results of the three participating systems are shown, plus our results for the method that we present in this paper. Our system did not participate in the SemEval task because it was developed afterwards. Our system missed some frames due to low-level tokenization and format issues, since our system was not designed specifically for the SemEval task.

Text	System	Precision	Recall	F-measure
Dublin				
	* Our system	0.7070	0.3162	0.4370
	CLR	0.6469	0.3984	0.4931
	LTH	0.7156	0.5184	0.6012
	UTD	0.7716	0.4188	0.5430
China				
	* Our system	0.6401	0.4261	0.5116
	CLR	0.6302	0.4621	0.5332
	LTH	0.7731	0.6261	0.6918
	UTD	0.8009	0.5498	0.6457
Work				
	* Our system	0.7336	0.4132	0.5286
	CLR	0.7452	0.5054	0.6023
	LTH	0.8642	0.6606	0.7488
	UTD	0.8382	0.5251	0.6457

Table 5. Frame element assignment results on the SemEval 2007 Task 19 test data. Only two of the participating systems worked on this task (CLR worked only on the previous task, frame assignment). Our system had the highest precision on the task of assigning frame elements.

Text	System	Precision	Recall	F-measure
Dublin				
	* Our system	0.63507	0.22027	0.32710
	LTH	0.54857	0.36345	0.43722
	UTD	0.53432	0.26238	0.35194
China				
	* Our system	0.56323	0.26245	0.35806
	LTH	0.57410	0.40995	0.47833
	UTD	0.53145	0.31489	0.39546
Work				
	* Our system	0.71053	0.28000	0.40170
	LTH	0.67352	0.30641	0.54644
	UTD	0.61842	0.45970	0.40978

References

1. Narayanan, S., Harabagiu, S.: Question and Answering based on semantic structures. In: Proceeding of the 20th International Conference on Computational Linguistics (COLING), pp. 693-701. (2004)
2. Boas, H.C.: Bilingual FrameNet dictionaries from machine translation. In: the proceedings of the Third International Conference on Language Resources and Evaluation (LREC), pp. 1364-1371. (2002)
3. Melli, G., Wang, Y., Liu, Y., Kashani, M.M., Shi, Z., Glu, B., Sarkar, A., Popowich, F. Description of SQUASH, the SFU question and answering summary handler for the DUC-2005 Summarization Task. In: Proceedings of the HLT/EMNLP Document Understanding Conference (DUC). (2005)
4. Fillmore, C.J.: Frame semantics and the nature of language. In: Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, vol.280 pp. 20-32 (1976)
5. Fillmore, C.J., Ruppenhofer, J., Baker, C.: FrameNet and representing the ink between semantic and syntactic relations. In: Huan, C., Lenders, W. (eds.) Frontiers in Linguistics, volume I of Language and Linguistics Monograph Series B. pp. 19-59. (2004)
6. Palmer, M., Kingsbury, P., Gildea, D.: The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics, vol. 31, pp. 71–106. (2005)
7. Kipper, K., Dang, H.T., Palmer, M.: Class based construction of a verb lexicon. In: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000). (2000)
8. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Computational Linguistics, vol. 28, pp. 245-288. (2002)
9. Baker, C., Ellsworth, M., Erk, K.: SemEval-2007 Task 19: Frame Semantic Structure Extraction. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pp. 99-104. (2007)
10. Bejan, C.A., Hathaway, C.:. UTD-SRL: A Pipeline Architecture for Extracting Frame Semantic Structures. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). (2007)
11. Florian, R., Cucerzan, S., Schafer, C., Yarowsky, D.: Combining classifiers for word sense disambiguation. In Natural Language Engineering. (2002)
12. Johansson, R., Nugues, P.: LTH: Semantic Structure extraction using nonprojective dependency trees. In: Proceeding of the 17th Inter-national Workshop on Semantic Evaluations (SemEval-2007). pp. 227-230. (2007)
13. Fellbaum, C. (ed.): WordNet: An electronic lexical database. MIT Press. (1998)
14. Litkowski, K.: CLR: Integration of FrameNet in a Text Representation System. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). pp. 113-116. (2007)
15. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann (ed.) (2005)
16. Järvinen, T.: Multi-layered annotation scheme for treebank annotation. In: Nivre, J., Hinrichs, E. (eds.), TLT 2003. Proceedings of the Second Workshop on Treebanks and Linguistic Theories, pages 93 – 104 (2003)
17. Shi, L., Mihalcea, R.: Semantic Parsing Using FrameNet and WordNet. In: Proceedings of the Human Language Technology Conference, companion volume (HLT/NAACL 2004), (2004)