Macrothink Institute™

International Journal of Linguistics
ISSN 1948-5425
2011, Vol. 3, No. 1: E9

Machine Learning for Automatic Labeling of Frames and Frame Elements in Text

Martin Scaiano (Corresponding author)

School of Electrical Engineering and Computer Science

University of Ottawa

Ottawa, ON, K1N 6N5, Canada

Tel: 1-613-562-5800 E-mail: mscai056@uottawa.ca

Diana Inkpen

School of Electrical Engineering and Computer Science

University of Ottawa

Ottawa, ON, K1N 6N5, Canada

Tel: 1-613-562-5800 E-mail: diana@site.uottawa.ca

Received: May 5, 2011 Accepted: May 30, 2011 doi:10.5296/ijl.v3i1.648

Abstract

The development of systems that extract a frame representation of text can lead to deeper semantics being used in natural language processing. We present the development of our system for extracting frames from text. Our system is trained on the FrameNet data and tested on the SemEval 2007: Task 19 Frame Extraction Task data. We use machine learning for labeling frames and frame elements, resulting in system with a good performance. We provide a detailed analysis of our methods, challenges, and results. We also provide enough details and analysis to allow other researchers to develop similar systems.

Keywords: Frame labeling, Frame element labeling, Semantic role labeling, Word sense disambiguation, Machine learning

1. Introduction

Recent years have seen the application of semantic role labeling in task such as question answering (Narayanan & Harabagiu, 2007), machine translation (Boas, 2002) (Wu & Fung, 2009), and summarization (Melli, Wang, Liu, Kashani, Shi, Glu, Sarkar, & Popowich, 2005). The application of these technologies has shown improved results in number of these tasks. Semantic role labeling (SRL) (Palmer, Gildea, & Xue, 2010) has been made possible by the creation of resources such as FrameNet (Fillmore, Ruppenhofer & Baker, 2004), Propbank (Palmer, Kingsbury, & Gildea, 2005), and VerbNet (Kipper, Dang, & Palmer, 2000). Furthermore tools, such as, Shalmaneser (Erk & Padó, 2006) (http://www.coli.uni-saarland.de/projects/salsa/shal/) and the LTH Semantic Parsing tools (Johansson & Nugues, 2007) (http://nlp.cs.lth.se/software/) have made labeling text with semantic roles more accessible.

Advancing natural language processing by using more detailed semantics is desirable. Frames provide good detailed types, which represent objects, ideas and events in a compact and semantically uniform way. Consider the following example which describes the same situation, but from different perspectives.

Example 1. Three perspectives on the same action

1. Jennifer bought candy from the store.
2. The store sold Jennifer candy.
3. Jennifer purchased candy from the store.

Each sentence in the Example 1 can be distilled down the one common frame, commercial_transaction. The roles of a frame are encoded as frame elements which similar to semantic roles. In the examples above the frame elements are identical, thus producing identical representations. Jennifer would be the buyer, the store would be the place (location) and the seller, and the candy would be the goods.

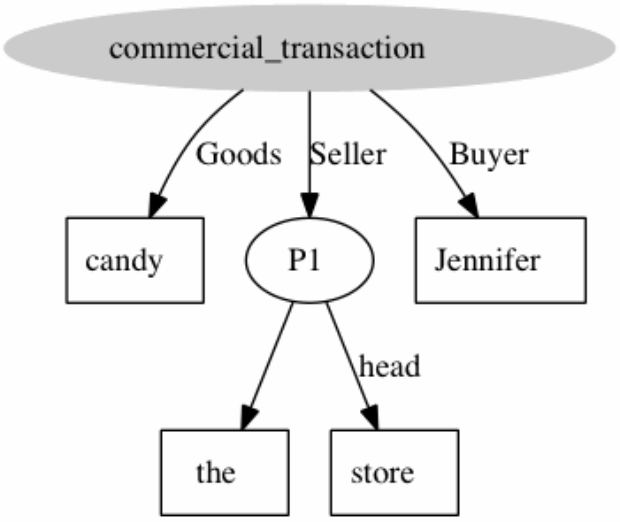

Figure 1. Uniform representation of frames for the sentences from Example 1

The grey ellipse at the top of Figure 1 represents the frame commercial_transaction. The rectangles represent words or syntactic structures, which are the values of frame elements. P1 is a syntactic structure that contains the words “the” and “store”.

In this article, we will discuss the development of our system for labeling frames and frame elements. Using our work as a guide, others could implement similar or better systems quickly.

1.1 Background

Since frames were first proposed by Minsky (1975), they have been actively researched for computer vision, natural language processing and automated reasoning. Frames are used to describe events, situations, or objects in a compact way. Frames assume a general understanding of the concepts, but the specific details of any instance are stored in frame elements, as in Figure 1. Frame elements are like parameters to a frame. Usually frame elements will assume certain default values or restrictions.

While frames can be a part of knowledge representation systems, usually representing concepts, frames do not define a whole representation and reasoning system. Conceptual Graphs (Sowa, 1984) offer a more complete representation system that is compatible with frames and are still actively researched.

1.2 FrameNet

To effectively use frames in any knowledge representation system, an inventory of frame definitions is required. FrameNet is currently the largest and best-known resource for English frame definitions. To some degree, frames are language-independent because the parameters of an object or situation are often language-independent; however, not all frames are language-independent, since they may have language-dependant implications. Many of FrameNet’s supporting resources are language-dependent.

While many different versions of FrameNet exist for different languages: Japanese (Ohara, Fujii, Saito, Ishizaki, Ohori, & Suzuki, 2003); Spanish (Carlos & Petruck, 2003); German (Boas, Ponvert, Guajardo, & Rao, 2006), none are as complete as Berkeley FrameNet, which provides a lexicon of English frames and examples. Even though there are a few other FrameNets available for different languages, we will refer to Berkeley FrameNet simply as FrameNet.

Some of the resources provided by FrameNet are definitions of frames, an ontology of frames, mappings between related frames, lists of lexical units (defined below) for each frame, and annotated texts exemplifying many frames and lexical units.

The current release of FrameNet, 1.3, contains definitions for 795 frames. These frame definitions are critical to building our actual representation. The list of defined frames is small, though it covers many common verbs and nouns.

Lexical units are the words or expressions that can evoke (bring to mind) a frame, for example: “buy” suggests a purchasing frame. Identifying when a frame is evoked from a set

of possible lexical units is a form of word sense disambiguation (Pedersen, 2002).

A word or expression may suggest multiple possible frames; for example, “bank”, may suggest a building for storing money, the side of river, to lean something sideways, or to bet on something (as in “don’t bank on it”). In some cases, even when one or more lexical units exist, none of them may represent the current situation. FrameNet 1.3 has about 10,000 lexical units. This does not cover much of the English vocabulary; but it does include many frequent words, which describe common situations.

The task of extracting frames from plain text was undertaken at SemEval 2007 as task 19: Frame Semantic Structure Extraction (Baker, Ellsworth, & Erk, 2007). Only three teams competed and only two completed the entire task. The task was divided into two parts: labeling of frames and labeling of frame elements.

CL Research, a participant in the SemEval 2007 Frame Extraction Task (Litkowski, 2007), integrated their frame extraction system into an already existing system. Their system used rules, a lexicon and additional processing to extract frames. Our method differs in that we focused on using machine learning with the FrameNet annotated corpus.

The UTD-SRL (Bejan & Hathaway, 2007) applied SVM and Maximum Entropy to the tasks of frame disambiguation and frame element labeling respectively. Their system used established methods for disambiguation (Florian, Cucerzan, Schafer, & Yarowsky, 2002). When their system faced decisions with an inadequate amount of training data (5 sentences or less) the system randomly choose a frame.

The LTH system (Johansson & Nugues, 2007) used a dependency parser for the initial data structure, as our system does. Before doing disambiguation, their system applied a small set of handcrafted rules to filter out words that are unlikely to evoke frames, because their presence could degrade the performance of the system (in particular prepositions). Frame disambiguation was done using an SVM classifier. The authors have extended the collection of lexical units in FrameNet by using WordNet (Fellbaum, 1998). The addition of new lexical units benefited their system in the task. The frame element labeling was done by using the words immediately related in the dependency tree.

1.3 System Overview

Our system is simply designed: first we require a dependency parse tree from the Machinese Semantic Parser (Järvinen, 2003); second we try to identify all the concepts using machine learning classifiers; third we label all relationships using machine learning classifiers.

Figure 2 shows our system as the outer blue box. This outer blue box can be considered as the infrastructure for using all the components together. Contained within the infrastructure are Common Resources such as the APIs for access to FrameNet, WordNet, WEKA (Witten & Frank, 2011) (http://www.cs.waikato.ac.nz/ml/Weka/), and a data model for the Machinese Semantic Parser. The outer box also contains links to tasks, which can be run using the system. Some examples are evaluation on SemEval 2007, Entailment of two sentences (though we do not focus on Entailment or Visualization in this article), and a Visualization of

a processed sentence.

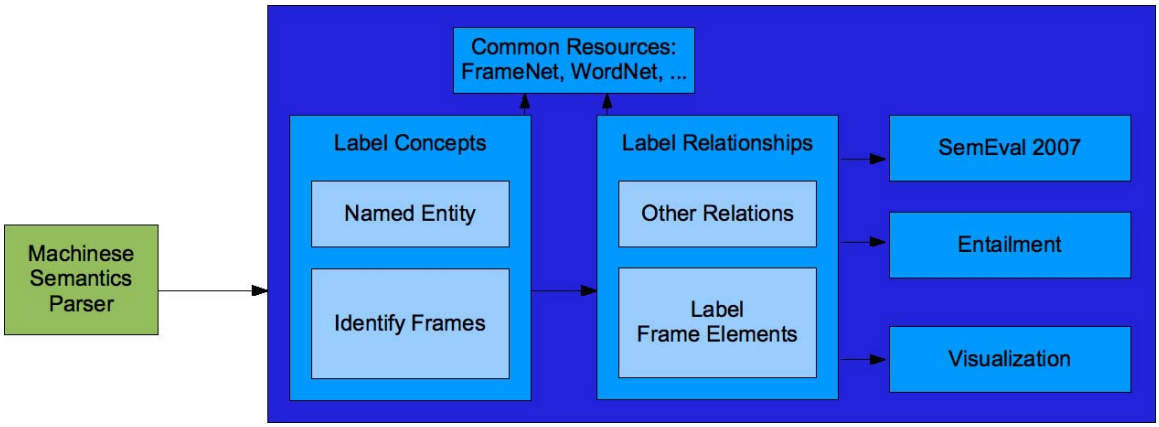

Figure 2. System overview

The Label Concepts box includes named entity recognition, identification of frames, evaluation of numbers, and recognition of other concepts. Since the focus of the article is labeling frames, we will only discuss the components and processing relevant to this work, which includes a few preprocessing rules, followed by named entity recognition and identifying frames.

The Label Relationships box includes processing for creating any relations between concepts, but we will focus here only on the frame elements, since other relationships are not used during labeling of the frame elements.

All major components of our system are needed to complete the evaluation on the SemEval 2007 task 19. This includes converting our representation into a compatible input representation and converting the output in the expected format.

1.4 Machinese Semantic Parser

The Machinese Semantics parser by Connexor Oy (Järvinen, 2003) produces a dependency tree enhanced with semantic information. The addition of semantic information such as the type of dependency relations, or the semantic class of nouns, should make the effort of classifying frame elements easier.

Machinese Semantics runs very quickly even on large sets of documents, and does not require an abundance of memory. The parser is excellent at identifying proper nouns, such as names and places, and some other general semantic information such as whether a noun is an animate, sentient, male/female, plant, or animal. The complete list of semantic information is limited, but still very useful.

The dependency relations provided by Machinese Semantics are excellent. They can accurately classify the type of relationship between two words.

The parser does suffer from one major failing: it frequently only finds partial parse trees. About 30% of our test sentences from the FrameNet annotated corpus could not be parsed into a complete dependency tree. Most of the incomplete parses present themselves with several roots, breaking some of the implied associations.

1.4 WEKA

Weka (Witten & Frank, 2011) is a java library that provides dozens of machine learning algorithms, including classifiers and clustering algorithms, and provides utilities to evaluate the resulting models.

Our system treats both the frame identification and the frame element labeling as classification tasks. There were a number of efforts made to optimize our usage of Weka. Memory usage and runtime were significant problems though many of these issues should become irrelevant as computer-processing power increases.

2. Methodology

Here we discuss a number of issues involved in our methodology including motivation, analysis, our decisions and alternatives, the methods themselves, and some key failed and successful experiments. We present much of our reasoning and analysis so that others may make optimal choices for their own goals. It should also be noted that we briefly discuss some performance and implementation issues in Appendix 1.

2.1 Initial Parsing

The system starts by parsing the text with the Machinese Semantic parser by Connexor Oy. The parser provides a dependency parse tree and limited semantic information about some words. The parser also tags named entities, again providing some limited semantic information, such as, if the name represents a country, city, or other location, and if it is a person the usual gender associated with the name. Dependency parse trees can increase the effectiveness and simplify the effort required to develop knowledge extraction systems (Johansson and Nugues, 2007).

2.2 Labeling Concepts

Our system labels concepts in two parts: there is a rule-based preprocessing and a frame identification step. The preprocessing includes named entity recognition; but for clarity we will discuss named entity recognition separately from the rest of the preprocessing because it is complex compared to all the other preprocessing rules.

2.3 Preprocessing

The preprocessing rules were initially derived from the rule described by Johansson and Nugues (2007). All of our preprocessing is based on examining the information provided by the dependency parse tree. We have added some rules to support a more complete knowledge representation system; our system now uses Conceptual Graphs (Sowa, 1984) for knowledge representation and frames are used (when appropriate) as concept types in our representation system.

Below are most of our preprocessing rules. The few rules not presented here are related to joining conjunctions and disjunctions, and other minor representation adjustments.

1. Do not consider any word that is not an adjective, adverb, noun, verb, pronoun or numeral as a possible frame. Some frames are evoked by prepositions; these are ignored because they are rarely relevant.
2. Do not consider words in verb chains such as modal modifiers (such as “would”, “should”, “could”). Verb chains are a specific relation in the dependency tree provided by the Machinese Semantic parser.
3. Do not process negation words, which can be identified by a particular dependency relation. Negation is not considered a part of frame identification, though it may be contained in a frame element.
4. Do not process the word “there” when it is in the dependency relation theme. The word “there” most often occurs in situations where it is simply indicating the existence of something. For example, “There is a bear that chases campers.”
5. Do not process the verb “be” as a frame. There is only one frame for the verb “be”: Performers_and_roles. The Performers_and_roles frame is used to indicate a person who is acting a particular role in a performance, such as a play. In most cases the verb “be” functions to indicate a co-reference relationship or a characteristic relationship.
6. Any word that is identified as a number by the parser, WordNet, or a dictionary, is processed specially. For the SemEval 2007 task, this means creating a Cardinal_number frame for the headword in a sequence of words representing a number.

2.4 Named Entity Recognition

The Connexor Semantic Parser already includes a named entity recognizer. The built-in named entity recognizer is quite effective at identifying names. Furthermore it seems to be enhanced with a simple knowledge base, which allows for a number of names to be classified into types such as country, city, male, or female. While these are general classes and seem to use the most frequent meaning of each word, they are still effective.

We tested classifying named entities using WordNet instead of relying on the parser and the result showed minimal difference. Our system uses first the type provided by the parser, but if a general type is found, we attempt to identify the type using WordNet.

When using WordNet to identify type classes for named entities, we selected a small number of key synsets positioned high up in the ontology that are likely to represent particular classes. For example, for location, city, country and person we use the first and most common synset associated with each those words. Sometimes extra synsets are needed to effectively represent the same concept. For example: location can also be represented by the following gloss, which is from land, not from location: the solid part of the earth's surface; "the plane turned away from the sea and moved back over land"; "the earth shook for several minutes"; "he dropped the logs on the ground".

When presented with a word identified as a named entity, but without specific type classification, it may be classified by testing how it relates to any of the key synsets

mentioned above. In particular we use the hypernym and instanceOf relation in WordNet. Each test is slightly different. Table 1 shows the word classes that we tested for using WordNet and different test methods.

Table 1. Test methods of word classification using WordNet

Class	Description of Test Method
Location	Is the named entity or any hypernym of it, an "instance of" something that is a hypernym of one of the key location synsets. For a location to be of interest it must be an instance of something that is a location.
Country	The country test is the same as the location test except that key synsets are for countries.
City	The city test is the same as the location test except that key synsets are for cities.
Person	Is the named entity a hypernym of the person key synset.
Animal	The animal test is the same as the person test except that the key synset is for animal.

2.5 Identifying Frames

The process of identifying which frames are evoked from our inventory is effectively a form of word sense disambiguation. Recent years have seen extensive research in word sense disambiguation using WordNet yet most methods achieve performance barely above the baseline of the most frequent sense (which tends to be quite high).

The LTH system also used an SVM classifier, following the method of Erk (2005), and used the following feature target lemma, target word, sub categorization frame (verb only), dependencies of the target word, the list of child words and the parent word in the dependency tree structure. We tested this same set of features, as is described below.

The baseline accuracy for this task is 74%, when only considering ambiguous examples and choosing the most frequent sense, when tested on FrameNet data, by cross-validation. This means training and testing on the same kind of data, and provides optimistic results. If this baseline method is applied to another corpus, for example the SemEval 2007 training corpus, the results are much worse (approximately 20% precision and recall). This is likely because the true frequencies are context dependant.

2.6 Training Data

The FrameNet project has been built up based on corpus evidence. Most of the frames and frame elements in FrameNet have been motivated and annotated in their corpus. The corpus provides annotated examples of frame and frame element realizations. This data can be used to train classifiers to disambiguate between several different frames.

The SemEval 2007 task 19 provided similar training data with a few new frames added to the annotations. Furthermore, the training data in SemEval 2007 was fully annotated; thus

meaning it could be used for negative training (training a classifier to know when a word does not evoke any known frame).

We used the SemEval 2007 task 19 test data as a held-out test set. The test data consists of three texts: Dublin, Work, and China. When optimization or analysis was needed, we only considered the Dublin text, while for final evaluation we would consider all the texts.

2.7 Classifier Configuration

We considered three main classifiers, Naïve Bayes, Decision Trees (J48) and SVM (SMO); all are from Weka and they are using default parameters. Naïve Bayes makes a good simple baseline classifier, J48 should be representative of rule-based methods and allows us to inspect what did the classifier learn, and SMO is a multi-class implementation of SVM, which is popular for this type of tasks.

To identify which words evoke frames we would simply apply a classifier to each word. We created one classifier per lemmatized headword for all lexical units. Thus any single classifier could only distinguish for a particular lemma, whether it evoked a frame and which frame.

While we did run a number of tests using negative training data from SemEval 2007, we found that the results improved precision, but significantly reduced recall. We choose not to use negative training data because the reduction in recall would hurt our comparison with other systems.

2.7.1 Classifier Features

The classifiers used information about the word and its context to determine the intended meaning. We found three simple features to be most effective for us:

1. The dependency relation of the word - This feature was the most effective. Intuitively if the relation is location, then the word should evoke a frame that is a location, likewise an agent relation should evoke a person, organization, country or some other sentient entity, or if the dependency relation is cause, then one would expect a frame the evokes an event.
2. A list of Boolean values indicating which lexical units are valid – This feature was intended to help with multi-word lexical units, by indicating whether the whole of the expression is present or not.
3. A list of Boolean values indicating which child dependencies are present for the word – This feature was to provide some context to the word, without using a bag of words. A bag of words seemed inappropriate since we have very limited training data for each frame and even less for each lexical unit. This feature provided an absolute increase in recall of 10% during the cross-validation experiments.

Table 2 shows the results of both cross-validation on the training data, and on the held-out test set. The cross-validated results are as usually optimistic compared to the held-out test set. In particular the large differences in recall were due to the presence in the test data of frames that do not exist in the training data. Table 2 shows the results of our first attempt to identify frames. We show the cross-validated results and the results on a hold out set.

Table 2. Frame identification intermediate results

Test		Cross-Validated		SemEval 2007 (Dublin)	
	Measure	Recall	Precision	Recall	Precision
Naïve Bayes		75%	60%	33%	61%
J48		75%	62%	33%	61%
SVM		75%	62%	32%	63%

2.7.2 Frame Identification Training Data Imbalance

A common problem in machine learning is insufficient or imbalanced training data; this is a problem with the training data from FrameNet. The large difference in recall in Table 2 is caused by the difference in distribution of frames between training and test data. Many lexical units in FrameNet had fewer than 5 training examples. This imbalance biases the classifiers towards particular classes. Furthermore, some lexical units had no training examples. Often, imbalance issues are solved by oversampling or undersampling, but each has disadvantages. Neither oversampling nor undersampling is ideal with this dataset, as some lexical units have no training data.

To resolve this issue we looked back at our data; since the goal of any classifier is to identify the evoked frame, we assumed that all expressions (lexical units) that evoke a particular frame would have common properties. While this assumption is not completely founded, it improved our cross-validated results and our results on our held-out test set.

To create the training data for any given classifier, we collect all examples that evoke the frames relevant to it. We could then undersample the training data to provide a more balanced classifier.

Table 3 shows the results of Naïve Bayes for cross-validation, and on the SemEval 2007 Dublin set. The results are much improved. As with Table 2, all classifiers performed very similarly and we ultimately choose to use SVM for our system. Table 3 shows the results of identifying frames with data imbalance correction.

Table 3. Frame identification intermediate results

Test		Cross-Validated		SemEval 2007 (Dublin)	
	Measure	Recall	Precision	Recall	Precision
Naïve Bayes		96%	99%	38%	66%

2.8 Frame Element Labeling

Labeling frame elements is very much like semantic role labeling (SRL), except that in the case of frame elements the labels are frame specific, where SRL uses general labels.

To assign frame elements to a frame, we trained frame specific classifiers that would assign a label from the list of frame elements of that frame or no frame element. Each classifier would consider words near the expression that evoked the frame and assign a label.

Table 4 shows the distribution of frame elements by relative position in the dependency tree from the frame evoking expression. Positive examples are examples of frame elements, while negative examples are relations that do not represent frame elements.

Table 4 covers 99% of the annotated examples; the missing 1% of frame elements did not fit into any of these relative positions. It is also noteworthy that the sentences with the missing frame elements all had parsing difficulties, resulting in partial parses.

Table 4. Distribution of frame elements by relative position

	Grandparent	Parent	Sibling	Child	Grandchild	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

Table 4 shows a significant imbalance, where negative examples make up 75% of the data and positive examples make 25%. Furthermore, the positive examples are subdivided into smaller classes representing the different frame elements and these classes are also imbalanced. This imbalance led to over-fitting during our initial work; some of the symptoms were as follows:

- Addition or removal of any single feature would not change the classifier results.
- Significant changes in the feature set had little impact on results.
- Difference between cross-validated results and held-out set seemed unrelated. The best cross-validated classifiers (SVM, RIPPER) performed worst on the held-out set and vice versa, with Naïve Bayes performing best on the held-out set.

To mitigate the impact of over-fitting, we did not include the grandparent and grandchild relative positions. This removed 5% or 12,318 of positive examples and removed 40% or 301,253 negative examples. This brings the positive / negative class balance to 32% / 68%.

It should be noted that grandparent relations make up less than 1% of all positive examples, but have an imbalance of 96% negative examples, while grandchild relations make up about 4% of positive examples, and are 96% negative examples. By not including grandparent and grandchild relations, 5% of frame elements are not detectable, but this should be offset by the improvement in data balance.

2.8.1 Frame Element Features

We tested features that other researchers (Gildea & Jurafsky, 2002) in semantic role labeling found to be effective. In addition to the features that most other semantic role labeling researchers have used, we also added the dependency relations as features.

Table 5. Frame element features

Feature	**Description**
Lexical Unit Id	The id of the lexical unit being labeled
Relative Position	The relative position of the word being labeled with respect the frame evoking word in the dependency tree

Dependency Relation	These are dependency relation defined by the Machinese Semantics parser. Some examples are main, object, condition, location, and coordination.
Part of Speech	These are basic parts of speech, such as, verb, noun, adjective, adverb, and article
Leading Preposition	This is the preposition, which introduces a phrase. The preposition (in a Machinese Semantics parse) is a child of the headword for the expression.
Semantic Type	These are semantic types provided by Machinese Semantics; there are very few and we only used: human, male, female, plant, animal, and nationality.
Voice of Verb	This indicates voice of the verb (passive, active) if the word is a verb otherwise it is set to none.
Is Animate	Indicates if the word is an animate object.

Table 5 describes the list of features that we tested for frame element classification. Most of the features did not have significant impact on the results. The most effective features were the dependency relation of the word being labeled, followed by relative location of the word.

2.8.2 Frame Element Classification Results

The results presented in Table 6 are the 10-fold cross-validation results for frame element labeling. The table shows the results of the three classifiers that we consistently tested. On a few occasions, we tested other classifiers and consistently found that they provided no significant improvement over these classifiers, thus we limited most of our work to these three classifiers.

Naïve Bayes gives the highest recall and the best F-measure in this circumstance. Naïve Bayes is less likely to overfit and thus will likely be more robust on the held-out set.

Table 6. 10-fold cross-validation frame element classification results

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

Even after our efforts to improve data balance, there still seems to be some imbalance. An idea we considered but did not test was to use the frame mappings (conversions between frames in the ontology) to create more examples of frame elements. A super type could assume that examples of frame elements from a sub type could also be used as examples in the super type. With the extra examples, undersampling could be used to now improve balance.

There may also be a method of improving training for subclasses by assuming that any frame element of a subclass should in some way conform to the expectation of the super class. We

attempted to test this method once. We automatically selected a set of likely synsets for each frame element from WordNet, then required that any frame element that could be mapped as a subtype of a parent frame element must at least meet the parent frame element requirements (a form of selectional preference or selectional restriction). The results were a decrease in recall and no improvement in precision. We believe our failure was due to our automatic selection of the likely synsets.

2.9 Evaluation on SemEval 2007 Task 19 Data

The SemEval 2007 task 19 required the participants to label frame and frame elements in plaintext (using the FrameNet inventory). The task was evaluated on the three texts mentioned above, and the results of each participant were compared to a gold standard using a Perl script.

While this task occurred before our work was done, it has provided a good held-out test set for our work. This held-out set varies significantly in the distribution of frames and frame element when compared with the FrameNet training data. Thus our results are lower than our cross-validation results, but provide an evaluation of the robustness of the system.

There were a number of challenges when using the published evaluation script. The following list highlights the issues; subsequent sections will discuss each issue and its resolution.

- Use of names instead of ids to denote frames and frame elements.
- Mismatching frames and frame element names.
- Evaluation script contained a divide by zero error.
- Differences in tokenization methods, thus comparison of tokens sometimes failed.
- Differences in the selection of frame element boundaries.

2.9.1 Mismatching Names Used Instead of Numeric Identifiers

FrameNet's annotated data has a small number of inconsistencies in the naming of frames and frame elements. Most of the inconsistencies are either capitalizations, or spaces versus underscores. A very small number of the issues are annotations for frame elements that do not exist for the particular frame.

Our primary resolution was to normalize all frame and frame element names in both the annotated examples and in the frame inventory. Normalization involved consistently using spaces instead of underscores and all names are in lower case letters. Any annotated examples that could not be corrected through normalization were manually corrected when possible, otherwise the annotation was ignored.

The comparison of names in the evaluation script was case sensitive and was not accepting of our normalized names, thus we modified the script to use a case insensitive evaluation.

These issues could have been avoided if the FrameNet and SemEval data had used the numeric identifiers for frame and frame elements instead of the names.

2.9.2 Evaluation Script Contained a Divide by Zero Error

The evaluation script contained a divide by zero error when there were no frames or frame elements to retrieve. The precision and recall are calculated both per sentence and for the overall text; thus if a single sentence caused this divide by zero error then the entire evaluation would fail. To correct this issue the evaluation script was modified, not to divide by zero, but to give a default value for precision and recall in these circumstances.

2.9.3 Differences in Tokenization Methods and Frame Element Boundaries

The difference in tokenization and frame element boundaries stem from our parser: the parser has it own tokenization which had to be mapped to the tokenization required for the SemEval 2007 task; our frame element boundaries were derived from the dependency tree which differ somewhat from the syntactic parses usually used for this task.

Our resolution to this problem was to map tokens from our dependency parser to the tokens that the SemEval task needed. We determined by manual evaluation that our results would only have an absolute increase of 1.5% if all tokens had been properly mapped or the frame element boundaries had not been slightly skewed. A difference of 1.5% does not significantly impact this evaluation.

3. Comparison of Related work on Evaluation on SemEval 2007 task 19

Table 7 lists the results of our system and the participants' systems (Baker, Ellsworth, & Erk, 2007) in SemEval 2007 task of frame identification. Our system is listed as the uOttawa. While this system consistently performs the weakest, it is only by a small margin and the simplicity of the design and implementation of our system makes it easily reproducible. The best frame labeling results were achieved by using the SVM classifier.

Table 7. Precision, recall, and F-measure of frame identification systems using the SemEval 2007 task 19 test data

Text	System	Recall	Precision	F-Measure
Dublin	uOttawa	0.3718	0.6446	0.4716
	UTD-SRL	0.4188	0.7716	0.5430
	LTH	0.5184	0.7156	0.6012
	CLR	0.3984	0.6469	0.4931
China	uOttawa	0.4261	0.6401	0.5116
	UTD-SRL	0.5498	0.8009	0.6457
	LTH	0.6261	0.7731	0.6918
	CLR	0.4621	0.6302	0.5332
Work	uOttawa	0.5000	0.7272	0.5926
	UTD-SRL	0.5251	0.8382	0.6457
	LTH	0.6606	0.8642	0.7488
	CLR	0.5054	0.7452	0.6023

Table 8 shows the results for the frame element labeling task; in fact these results represent the combined frame identification and frame element labeling (any errors in the first task impact the final performance). For frame element labeling we used the Naïve Bayes classifier. Our system shows competitive results of combined frame element labeling and frame labeling. These results show our best balance between precision and recall. There are a number of different configurations, which provided much higher precision at the expense of recall; for example, the previously mentioned attempt to use WordNet to identify likely synsets for particular frame elements.

Table 8. Precision, recall, and F-measure of joint frame and frame element identification systems using the SemEval 2007 task 19 test data

Text	System	Recall	Precision	F-Measure
Dublin	uOttawa	0.30356	0.49642	0.37674
	UTD-SRL	0.26238	0.53432	0.35194
	LTH	0.36345	0.54857	0.43722
China	uOttawa	0.36756	0.51754	0.42984
	UTD-SRL	0.31489	0.53145	0.39546
	LTH	0.40995	0.57410	0.47833
Work	uOttawa	0.35704	0.57933	0.44180
	UTD-SRL	0.30641	0.61842	0.40978
	LTH	0.45970	0.67352	0.54644

4. Conclusions

We have built and refined a system for labeling frame and frame elements, which is effectively a word sense disambiguator and a semantic role labeler, with a very simple design, which can quickly and easily be reproduced. The results of the system are comparable to other systems, yet they leave much room for improvement.

The uses of the Machinese Semantic Parser was an excellent foundation to our system, though it could be replaced with any good dependency parser, by also including a named entity recognizer, and by using WordNet for semantic classification.

In this article, we documented a number of our successful efforts, our analysis, and our failed efforts, so that other researchers may improve upon our work. We have even presented a number of ideas, which we have not been able to explore.

Future work based on this system will likely involve testing it on tasks such as entailment, question answering, novelty detection (detection of novel or new information) and other reasoning-based tasks.

References

Baker, C., Ellsworth, M., & Erk, K. (2007). SemEval-2007 Task 19: Frame Semantic Structure Extraction. *Proceedings of the Fourth International Workshop on Semantic Evaluations* (SemEval-2007), ACL 2007, pages 99-104

Bejan, C.A., & Hathaway, C. (2007). UTD-SRL: A Pipeline Architecture for Extracting Frame Semantic Structures. *Proceedings of the Fourth International Workshop on Semantic Evaluations* (SemEval-2007), ACL 2007

Boas, H.C. (2002). Bilingual FrameNet dictionaries from machine translation. *Proceedings of the Third International Conference on Language Resources and Evaluation* (LREC), pp. 1364-1371

Boas, H.C., Ponvert, E., Guajardo, M., & Rao, S. (2006). *The current status of German FrameNet*. SALSA workshop at the University of the Saarland. Saarbrücken, Germany

Erk, K. & Padó, S. (2006). Shalmaneser - a flexible toolbox for semantic role assignment. *Proceedings of LREC-2006*, Genoa, Italy. [Online] Available: http://www.coli.uni-saarland.de/projects/salsa/shal/

Erk, K.. (2005). Frame assignment as word sense disambiguation. *Proceedings of IWCS 6*.

Fellbaum, C. (editor). (1998). WordNet: An electronic lexical database. MIT Press

Fillmore, C.J., Ruppenhofer, J., & Baker, C. (2004). FrameNet and representing the link between semantic and syntactic relations. Churen Huan and Winfried Lenders, editors, *Frontiers in Linguistics*, volume I of Language and Linguistics Monograph Series B. Institute of Linguistics, Academia Sinica, Taipei, pages 19-59

Florian, R., Cucerzan, S., Schafer, C., & Yarowsky, D. (2002). Combining classifiers for word sense disambiguation. *Natural Language Engineering*

Gildea, D., & Jurafsky, D., (2002). Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3): pages 245-288

Järvinen, T. (2003). Multi-layered annotation scheme for treebank annotation. Joakim Nivre and Erhard Hinrichs, editors, TLT 2003. *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pages 93 -104. Växjö University Press

Johansson, R. & Nugues, P. (2007). LTH: Semantic Structure extraction using nonprojective dependency trees. In Proceeding of the 17th Inter-national Workshop on Semantic Evaluations (SemEval-2007), pages 227-230, Prague, Czech Republic. [Online] Available: http://nlp.cs.lth.se/

Kipper, K., Dang, H.T., & Palmer, M. (2000). Class based construction of a verb lexicon. *Proceedings of the 17th National Conference on Artificial Intelligence* (AAAI-2000), Austin, TX

Litkowski, K. (2007). CLR: Integration of FrameNet in a Text Representation System. *Proceedings of the Fourth International Workshop on Semantic Evaluations* (SemEval-2007), pages 113-116, ACL 2007

Melli, G., Wang, Y., Liu, Y., Kashani, M.M., Shi, Z., Glu, B., Sarkar, A., & Popowich, F. (2005). Description of SQUASH, the SFU question and answering summary handler for the DUC-2005 Summarization Task. *Proceedings of the HLT/EMNLP Document Understanding*

Conference (DUC), Vancover, Canada

Minsky, M. (1975). A Framework for Representing Knowledge. In P. H. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill, pp. 211-277

Narayanan, M. & Harabagiu, S. (2007). Question Answering based on semantic structures. *Proceeding of the 20th International Conference on Computational Linguistics* (COLING), pages 693-701, Geneva, Switzerland

Ohara, K.H., Fujii, S., Saito, H., Ishizaki, S., Ohori, T., & Suzuki, R. (2003). The Japanese FrameNet Project: A Preliminary Report. *Proceedings of Pacific Association for Computational Linguistics* (PACLING'03), pages 249-254. Halifax, Canada

Palmer, M., Gildea, D., & Xue, N. (2010). Semantic Role Labeling. *Human Language Technologies*

Palmer, M., Kingsbury, P., & Gildea, D. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31, 71–106

Pedersen, T. (2002). A baseline methodology for word sense disambiguation. *Computer Science*

Sowa, J.F. (1984). Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading, MA

Subirats, C. & Petruck, M. (2003). Surprise: Spanish FrameNet. Presentation at Workshop on Frame Semantics. *International Congress of Linguists*. Prague, Czech Republic

Witten, I.H. & Frank, E. (2011). Data Mining: Practical machine learning tools and techniques, 3rd Edition. San Francisco: Morgan Kaufmann

Wu, D., & Fung, P. (2009). Can Semantic Role Labeling Improve SMT? [Online] Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.157.6546

Glossary

LTH: Lunds Tekniska Högskola – The Faculty of Engineering at Lund University in Sweden

SRL: Semantic Role Labeling – the task of assign roles to syntactic arguments and adjuncts

SVM: Support Vector Machine – A machine-learning algorithm

UTD-SRL: University of Texas at Dallas Semantic Role Labeling system – A system entered into the SemEval 2007 task 19.

WEKA: Waikato Environment for Knowledge Analysis – A java application, library, and toolkit for machine learning problems containing several algorithms

Appendix

Appendix 1. Performance Notes

While developing the system, a number of performance issues were encountered and we would like to quickly comment on them. Early in system's development, training the classifiers could take about a week; now training takes only a few hours.

To minimize Weka's training time on classifiers such as J48 and SVM, arff files should be minimized; arff files should be preprocessed to remove any unused nominals, for example, dependency relations which never appeared in a particular arff file.

All models in Weka should be saved using the gzipped (compressed) format, because the uncompressed models use much more disk space and they also take much longer to load. Any large data sets, measured in gigabytes, tend to be accessed faster if they are compressed. That is to say, on-the-fly decompression is often faster than reading a decompressed file, particularly if it is plain text.

A minimum of 8 GB of memory was needed to train many of the WEKA classifiers.