

CLR: Linking Events and Their Participants in Discourse Using a Comprehensive FrameNet Dictionary

Ken Litkowski
CL Research
Damascus, MD USA.
ken@clres.com

Abstract

The CL Research system for SemEval-2 Task 10 for linking events and their participants in discourse is an exploration of the use of a specially created FrameNet dictionary that captures all FrameNet information about frames, lexical units, and frame-to-frame relations. This system is embedded in a specially designed interface, the Linguistic Task Analyzer. The implementation of this system was quite minimal at the time of submission, allowing only an initial completion of the role recognition and labeling task, with recall of 0.112, precision of 0.670, and F-score of 0.192. We describe the design of the system and the continuing efforts to determine how much of this task can be performed with the available lexical resources. Changes since the official submission have improved the F-score to 0.266.

1 Introduction

The semantic role labeling (SRL) task has received considerable attention in recent years, with previous tasks in Senseval-2 (Litkowski, 2004), Semeval-1 (Baker et al., 2007), and CoNLL (Carreras & Marquez, 2004; Carreras & Marquez, 2005). The current task, Linking Events and their Participants in Discourse, continues the evolution of SRL tasks with the intent of identifying Null Instantiations, i.e., frame elements that are absent from the local context, but potentially recoverable from the wider discourse context.

CL Research participated in one subtask, role recognition and labeling, unable to implement techniques for the null instantiation subtask. This paper describes our efforts thus far (clearly a work in progress), specifically the implementation of a development interface (section 2), the use of a specially constructed FrameNet dictio-

nary (section 3), techniques for performing the role recognition and labeling task (section 4), our results (section 5), and future developments (section 6).

2 The Linguistic Task Analyzer

CL Research participated in the linking task by extending its Linguistic Task Analyzer (LTA), an interface also used for such tasks as word-sense disambiguation and recognizing textual entailment. LTA includes a wide array of modules, including a full-scale parser, post-parsing semantic analysis routines, the use of XML functionality for creating and analyzing input and output, and access to several integrated dictionaries (used for semantic analysis). Modification of LTA for the linking task involves using existing functionality and implementing new functionality specific to the task. We describe LTA in some detail to illustrate steps that might be relevant to a symbolic approach to the linking task.

Each task in LTA consists of a set of items to be analyzed, in this case, an identifier for each sentence in the document being analyzed. LTA loads the appropriate XML files (usually the annotation file and the gold file) and provides various data for each sentence, including the number of terminals, non-terminals, frames, frame elements that have been recognized, true positives, false positives, false negatives, and a characterization of problems that have been encountered. Summary statistics are given, showing such things as the total number of frames and the scoring for the current annotation (when a gold file is available).

Whenever a sentence is selected in the LTA, the text is shown (accomplished by querying the XML for the selected sentence and retrieving all its terminals). LTA provides a capability for se-

lecting all sentences matching particular criteria, e.g., all sentences containing a **Color** frame or all sentences having targets that have problematic entries in the FrameNet dictionary.

LTA contains a basic command to run and evaluate the system against the selected sentences. This can be used during development to test the effect of changes to the underlying code for performing any of the tasks. During the test phase, all sentences are selected, the **Run and Evaluate** command is executed, the XML test file is modified with the insertion of frame elements constituting the system's answers, and the XML file is saved for the official submission. For the official submission, this took less than a minute for each of the two chapters.

A single sentence can be selected in the LTA for detailed examination. This **Sentence Detail** shows (1) the sentence itself (as in the main form), (2) a tree of the frames in the sentence, along with each of the frame elements that have been identified, minimally showing the target, and the text that has been identified for the frame element, and (3) from the training data, the frame element differences from the gold file, along with their terminal or non-terminal id references.

The **Sentence Detail** also has buttons to (1) score the annotation against the gold file for the sentence, (2) identify the missing core frame elements, (3) examine the FrameNet entries for the targets, and (4) perform the task. The functionality underlying the scoring and the task performance are called from the main form when all or selected sentences are to be processed (e.g., in the **Run and Evaluate** command).

Implementation of the scoring functionality for the **Sentence Detail** form attempts to follow the implementation in the official scorer. We have not yet captured every nuance of the scorer; however, we seem to have 99.9 percent agreement.

The **Sentence Detail** functionality is at the heart of the investigation and implementation of techniques for performing the tasks. At this time, we must view the implementation as only in its initial stages, minimally capable of performing the role recognition and labeling task. Further details about the implementation, including its shortcomings, will be described below.

3 The FrameNet Dictionary

Central to the performance of the linking task is the use of a dictionary constructed from the FrameNet data. This dictionary is in a format used

by the CL Research DIMAP dictionary maintenance program.¹ The FrameNet dictionary attempts to capture all the information in FrameNet, in a form that can be easily accessed and used for tasks such as the linking task. This dictionary is also used in general word-sense disambiguation tasks, when all words in a text are simultaneously disambiguated with several dictionaries. The FrameNet dictionary has almost 11,000 entries² of four main types: frames, frame-to-frame relations, normal entries, and frame elements³. This dictionary was initially described in Litkowski (2007), but is described in more detail in the following subsections in order to show how the information in these entries is used in the linking task.

3.1 Frame Entries

A FrameNet frame is entered in the dictionary by preceding its name with a “#” sign to distinguish it from other types of entries. A frame entry, such as **#Abandonment**, consists of one sense with no part of speech. This sense contains a list of its frame elements and the coreness of each frame element. The sense also lists all the lexical units associated with the frame, along with the identifying number for each so that a link can be made if necessary to the appropriate lexical unit and lexical entry XML files. The sense identifies any frame-to-frame relations in which the frame participates, such as “**IS_INHERITED_BY**” with a link to the inheriting frame. Thus, whenever a specific frame is signaled in the linking task, its properties can be accessed and we can investigate which of the frame elements might be present in the context.

3.2 Frame-to-Frame Relations

While the entries for the individual frames identify the frame-to-frame relations in which a frame participates, separate entries are created to

¹ These dictionaries are stored in a Btree file format for rapid access. A free demonstration version of DIMAP is available at CL Research (<http://www.cres.com>). This version can be used to manipulate any of several dictionaries that are also available. These include WordNet and the basic FrameNet. CL Research also makes available a publicly available FrameNet Explorer and a DIMAP Frame Element Hierarchy dictionary.

² By contrast, the DIMAP dictionary for WordNet contains 147,000 entries.

³ When a new version of FrameNet is made available, a new version of the DIMAP dictionary is created. This was the case with the preliminary FrameNet version 1.4a made available by the task organizers. This creation takes about two hours.

hold the mappings between the frame elements of the two frames. These entries are prefixed with an “@” sign, followed by the name of a frame, the frame relation, and the name of the second frame, as in the name “@Abounding_with INHERITS Locative_relation”. The single sense for such an entry shows the mapping, e.g., of the Location frame element of **Abounding_with** to the Figure frame element of **Locative_relation**. The information in these entries has not yet been used in the linking task.

3.3 Frame Elements

Frame element entries are preceded with a “%”, as in **%Toxic_substance**. We have a taxonomy of the 1131 uniquely-named frame elements in all the FrameNet frames.⁴ Each frame element entry identifies its superordinate frame element (or none for the 12 roots) and the frame elements in which it is used. The information in these entries has not yet been used in the linking task.

3.4 Main Entries

The bulk of the entries in the FrameNet dictionary are for the lexical units. An entry was created for each unique form, with senses for each lexical unit of the base form. Thus, **beat** has four senses, two verb, one noun, and one adjective. Minimally, each sense contains its part of speech, its frame, and its id number. A sense may also contain a definition and its source, if present in the FrameNet lexical unit files.

If available, the information available in the lexical entry (LE) files is encapsulated in the sense, from the FERealization elements. This captures the phrase type, the grammatical function, the frame element, and the frequency in the FrameNet annotation files. An example of what information is available for one verb sense of **beat** is shown in Table 1.

Table 1. Lexical Entry Syntactic Patterns for “beat”

Feature Name	Feature Value
NP(Ext)	Loser (12)
NP(Obj)	Loser (28)
PP[by](Dep)	Winner (5)
CNI()	Winner (5)
PP[against](Dep)	Winner (2)
NP(Ext)	Winner (31)

⁴ This taxonomy can be viewed at <http://www.cres.com/db/feindex.html>, which provides links describing how it was constructed and which can be downloaded in DIMAP or MySQL format.

At the present time, this type of information is the primary information used in the linking task.

4 Role Recognition and Labeling

To perform the role recognition and labeling task, the system first retrieves all the frames for the sentence and then iterates over each. The frame name and the target are retrieved. From the target XML, the id reference is used to retrieve the part of speech and lemma from the targets terminal node. With this information, an attempt is made to add child nodes to the frame node in the XML, thus supplying the system’s performance of the task. After any nodes have thus been added, it is only necessary to save the modified XML as the output file.

The first step in adding child nodes is to obtain the lexical entries from the FrameNet dictionary for the frame and the lemma. Since the lemma may have multiple senses, we obtain the specific sense that corresponds to the frame. We iterate through the features for the sense, focusing on those providing syntactic patterns, such as those in Table 1. We deconstruct the feature value into its frame element name and its frequency. We then call a function with the feature name and the target’s id reference to see if we can find a matching constituent; if successful, we create a child node of the frame with the frame element name and the id reference (for the child <fe-node> of frame element <fe> node).

The matching constituent function operates on the syntactic pattern, calling specific functions to search the XML terminals and non-terminals for constituent that fit the syntactic criterion. At present, this only operates on four patterns: **DEN()**, **Poss(Gen)**, **NP(Ext)**, and **N(Head)**.⁵ As an example, for **Poss(Gen)**, we select the non-terminals with the target as the “head” and search these for a terminal node marked as **PRP\$**. A special constituent matching function was also written to look for the **Supported** frame element in the **Support** frame.

5 System Results

CL Research’s results for the role recognition and labeling task are shown in Table 2. These results are generally consistent across the two chapters in the test and with results obtained with the training data during development. Combining

⁵ The DEN pattern identifies incorporated frame elements. Since the official submission, two patterns (**NP(OBJ)** and **PP(Dep)**) have been added.

the two chapters, the recall was 0.112, the precision was 0.670, and the F-score was 0.192.⁶

Table 2. Scores for Chapters 13 and 14

Measure	Ch. 13	Ch. 14
True Positives	191	246
False Positives	82	133
False Negatives	1587	1874
Correct Labels	189	237
Precision	0.700	0.649
Recall	0.107	0.116
F-Score	0.186	0.197
Label Accuracy	0.106	0.112

As can be seen, for entries with patterns (albeit a low recall), a substantial number of frame elements could be recognized with high precision from a very small number of constituent matching functions. A detailed analysis of the results, identifying the contribution of each pattern recognition and the problem of false positives, has not yet been completed. One such observation is that when the same syntactic pattern is present for more than one frame element, such as **NP(Ext)** for both **Loser** and **Winner** in the case of **beat** as shown in Table 1, the same constituent will be identified for both.

A significant shortcoming in the system occurs when there are no syntactic patterns available for a particular sense (27 percent of the targets). For example, the lemma **hour** frequently appears in the training set as the target of either the **Measure_duration** or **Calendric_unit** frames, but it has no syntactic patterns (i.e., the FrameNet data contain no annotations for this lexical unit), while **decade**, also used in the same frames, does have syntactic patterns. This is a frequent occurrence with the FrameNet dictionary.

6 Future Developments

As should be clear from the preceding description, there are many opportunities for improvement. First, several improvements can be made in the LTA to improve the ability to facilitate development. The LTA has only barely begun exploitation of the many integrated modules that are available. Additional functionality needs to be developed so that it will be possible to determine the effect of any changes in constituent matching, i.e., what is the effect on recall and

⁶The additional patterns described in the previous footnote have improved recall to 0.166 and F-score to 0.266, while maintaining a high precision (0.676).

precision. The sentence detail form can be improved to provide better insights into the relation between syntactic patterns and their matching constituents.

Secondly, major improvements appear likely from greater exploitation of the FrameNet dictionary. At present, no use is made of the frequency information or the weighting of choices for matching constituents. When a given lemma has no syntactic patterns, it is likely that some use of the patterns for other lexical units in the frame can be made. It is also possible that some general patterns can be discerned using the frame element taxonomy.

It is important to see how far the FrameNet data can be further exploited and where other lexical data, such as available in WordNet or in more traditional lexical databases, can be used. The data developed for this linking task provide many opportunities for further exploration.

References

- Collin Baker, Michael Ellsworth, and Katrin Erk. 2007. Semeval-2007 Task 19: Frame Semantic Structure Extraction. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, Association for Computational Linguistics, pp. 99-104.
- Xavier Carreras and Luis Marquez. 2004. Introduction to the CoNLL-2004 Shared Task Semantic Role Labeling. *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) International Workshop on Semantic Evaluations (SemEval-2007)*. Boston, MA Association for Computational Linguistics, pp. 89-97.
- Xavier Carreras and Luis Marquez. 2005. Introduction to the CoNLL-2005 Shared Task Semantic Role Labeling. *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) International Workshop on Semantic Evaluations (SemEval-2007)*. Ann Arbor, MI Association for Computational Linguistics, pp. 152-164.
- Kenneth C. Litkowski. 2004. Senseval-3 Task: Automatic Labeling of Semantic Roles. *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain, Association for Computational Linguistics, pp. 9-12.
- Kenneth C. Litkowski. 2007. CLR: Integration of FrameNet in a Text Representation System. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, Association for Computational Linguistics, pp. 113-6.