Proceedings of the Second Annual Conference on Advances in Cognitive Systems ACS-2013 (125-134)

Toward Learning High-Level Semantic Frames from Definitions

Jansen Orfan JORFAN@CS.ROCHESTER.EDU
James Allen JAMES@CS.ROCHESTER.EDU
Department of Computer Science, University of Rochester, Rochester, NY 14627 USA

Abstract

We describe work toward learning semantic frames from definitions in WordNet. Our ultimate purpose is to build a resource of common sense knowledge with at least the depth of FrameNet but with wider coverage. The approach uses the TRIPS parser to identify relationships between concepts which we evaluate with human judges. From those relations we generate a very small domain of frames with encouraging results. We discuss how the relationships can be further exploited in the future and identify obstacles and possible solutions.

1. Introduction

It is well recognized that effective cognitive systems must contain significant amounts of common-sense knowledge, and that it does not seem possible to encode such knowledge by hand. Rather, systems will have to learn such knowledge. One of the most promising approaches is to learn from reading, given the vast amount of textual information available. Here we report on work toward automatically deriving semantic frames, along the lines of those encoded in FrameNet (Johnson et al., 2003).

Resources like FrameNet provide knowledge about typical situations (frames), their potential participants (elements), and the words that evoke them (lexical units). Frame semantics can provide high-level knowledge to facilitate high-level cognition that requires an understanding of not just language but also situations in general. Furthermore, the symbolic nature of a frame allows researchers to better verify what a frame-based system knows and to understand entailments derived from it.

In practice, researchers mostly use FrameNet for not just shallow tasks like semantic parsing (Shi & Mihalcea, 2005) but also deeper ones like textual entailment (Burchardt et al., 2009). The caveat is that in databases like FrameNet, the entries must be created by expert users which has lead to a widely recognized coverage problem in its lexical units (Baker, 2012). There is also a coverage problem for less general frames. For example, FrameNet does not have a frame for any specific sports situations, like "strike" or "penalty kick"; instead there is a highly underspecified frame, Sports_Jargon, with only three lexical units (LU's), serve.n, strike(baseball).n, and strike(bowling).n. Previous attempts to create a lexical database of semantic frames with more coverage either ignore the deep knowledge aspects of frames or only focus on combining existing hand-built resources and do not address the problem of frame coverage.

In this paper we propose a novel approach that not only learns frames and the words that evoke them - with comparatively greater coverage than FrameNet - but also discovers semantic knowledge about how the evokers relate to their frames and how frames relate to each other. We have a secondary objective which is to classify those relations into abstract frame elements and inter-frame relations like those found in FrameNet. In the next section (2 Related Work) we will explore generalizations of previous work; afterwards, we will lay out our proposed approach (3 System) and present and discuss results (4 Experiments). Finally, we will briefly discuss future work and then conclude (5 Future Work and 6 Conclusion).

2. Related Work

Previous work related to our goal can be grouped into two categories based on both method and input source. The more trodden approach, variations of which appear in (Materna, 2012) and (O'Connor, 2012) uses statistical methods on large annotated corpora to learn the words associated with each verb - e.g. the frequency that the word, "man", occurs as the subject of the verb, "eat". Verbs are then clustered based on what words they co-occur with and each cluster forms a frame-like structure and the co-occurring words become that frame's evokers.

Statistical methods are very flexible in that they can use any number of large corpora as input which ensures their lexical unit coverage will be better than FrameNet's. They can also potentially employ hierarchical clustering that would allow hierarchical frames and very specific frames unlike FrameNet in certain cases like Sports_Jargon. Of course, these methods all come with a caveat: the corpora must contain sufficient information regarding a domain in order to learn anything about it. This is not a problem when the domain is often written about, like soccer, where the very specific concepts necessary to understanding the domain (passing, scoring, blocking, etc) are used repeatedly; however, by definition, obscure terms are unlikely to be learned. The depth of knowledge generated is an additional issue with existing statistical methodologies. FrameNet has a very specific and frame-dependent role set but (Materna, 2012) and (O'Connor, 2012) are only considering shallow grammatical relations like subject and object. The exclusion of a deep concept ontology and role hierarchy does not mean that statistical methods are inherently not capable of generating deeper knowledge but it will make the available data sparser as the general roles and concept are divided into more specific ones, e.g. the object role will be divided into theme, patient, etc.

The other type of method considered uses a more symbolic approach over hand built knowledge bases (KBs) like FrameNet, WordNet (Miller, 1995), and VerbNet. These methods attempt to combine the semantic richness of FrameNet with the coverage of machine readable dictionaries like WordNet by finding mappings between them. (Shi & Mihalcea, 2005) uses the mappings VerbNet provides to both WordNet senses and FrameNet frames to infer mappings directly from WordNet senses to frames. Furthermore, (Shi & Mihalcea, 2005) and (Laparra & Rigau, 2009) use shallow similarity measures (e.g. word overlap) on WordNet senses and LU glosses to directly map between the two. With the mappings in place, a frame's LU coverage can be expanded by incorporating their corresponding WordNet senses, synsets, and their hypernyms.

Learning from KBs has the advantage of incorporating deeper semantic knowledge - like WordNet's cause relations or FrameNet's preceded-by relations - that shallow statistical methods do not.

This comes at the cost of limited frame coverage - i.e. these methods only augment existing frames and semantic relations with another KB's knowledge and will not learn anything beyond what has been hand annotated. Compare this to a statistical method which can potentially learn a large breadth of, albeit shallow, knowledge.

3. System

Our goal is to learn new frames of a greater specificity like the statistical methods above but with an eye toward deeper semantic knowledge. Our approach works by using the deep representations of the content of word definitions from dictionaries described in (Allen et al., 2013) to derive frames and the concepts that evoke them. One of the benefits of using a dictionary is that they aim for wide coverage, so obscure word senses are likely to be encountered. We are using the definitions present in WordNet which has the added benefit that the frame evokers are also WordNet senses which provide a much deeper representation and potential for linking to other KB's. However, our process is not specific to WordNet and could potentially be used on any dictionary resource. The approach treats each WordNet sense as a potential frame (for this paper we are only concentrating on verbs) and searches for the other senses that use it in their own glosses - i.e. given a sense(frame), the system tries to find all of the senses(evokers) that evoke it.

We begin by producing an OWL 2 KB (the description logic subset of the web ontology language) from WordNet glosses using the TRIPS semantic parser as described in (Allen et al., 2013). In short, the parser produces a logical form (LF) from each gloss that represents a graph of nodes, their types (often WordNet senses but sometimes just types from TRIPS' upper ontology), and the semantic relations between them. For each sense we create an OWL2 class and then assert that it is a subclass of its gloss' LF. The LF's are converted to OWL2 in a fairly direct way, each node is translated into a new class which is subsumed by its type. Edges in the LF become existentially quantified property restrictions (although the choice of quantifier does not play a significant role for this paper's purpose). We also assert that every sense in a synset is equivalent. Notice that we do not use WordNet's hypernym hierarchy and all subsumption relations come from the glosses - this helps make the approach portable to non-hierarchical dictionary resources.

Figure 1 shows an LF graph for the gloss of pesticide%1:27:00, "a chemical used to kill pests". Each node lists its unique identifier and then its type's unique identifier. The source of the graph, "DEFINITION" points to the node that will become pesticide%1:27:00's subsumer. Reading the graph, we see that the defined concept is a chemical%1:27:00 such that it is applied (apply%2:34:00) for the purpose of killing (kill%2:35:00) pests (pest%1:05:00). From this LF we generate the following OWL2 axioms: **pesticide%1:27:00** $\sqsubseteq$ **N1**; **N1** $\sqsubseteq$ **chemical%1:27:00** $\sqcap \exists theme^{-}.$**N2**;

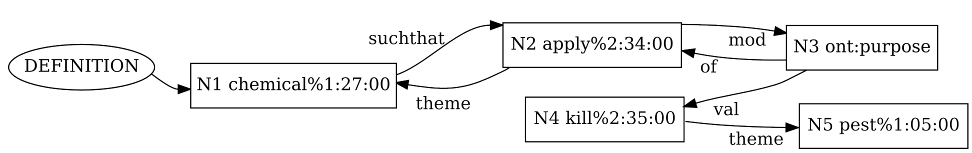

Figure 1. LF for, "a chemical used to kill pests"

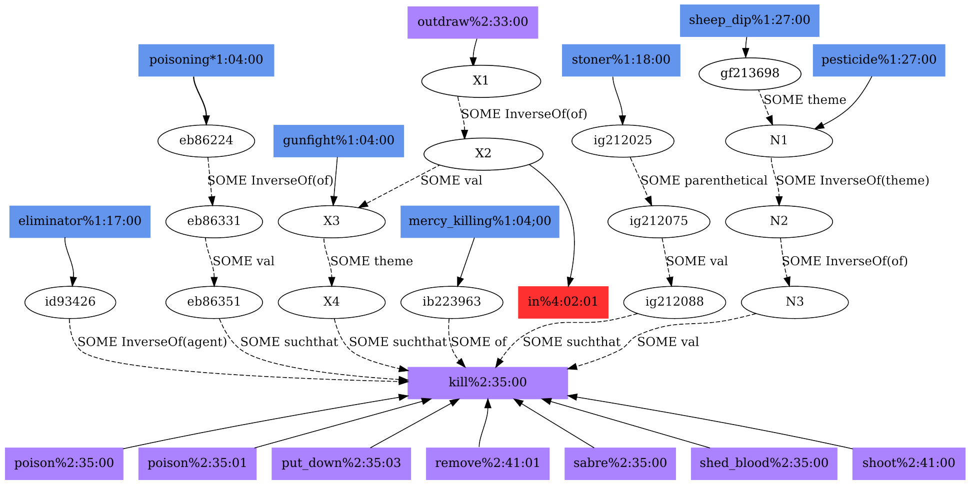

Figure 2. Subgraph of kill%2:35:00's evokers and the entailed semantic relations between them. Solid lines denote entailment and broken lines denote a quantified restriction. The graph shows simple relations like shoot%2:41:00 entailing kill%2:35:00 and an eliminator%1:17:00 being the agent of kill%2:35:00. It also shows more complex relations like pesticides%1:27:00 being a type of chemical(N1) that is used(N2) to(N3) kill(kill%2:35:00) and that outdraw%2:33:00 means to "best(X1) someone in(X2) a gunfight%1:04:00(X4)" and a gunfight%1:04:00 "involves(X4) guns (X5) associated with killing (kill%2:35:00)".

N2 $\sqsubseteq$ **apply%2:34:00** $\sqcap\ \exists of^{-}$.**N3**; **N3** $\sqsubseteq$ **ont:purpose** $\sqcap\ val$.**N4**; **N4** $\sqsubseteq$ **kill%2:35:00** $\sqcap$ *theme*.**N5**; **N5** $\sqsubseteq$ **pest%1:05:00**.

We then run an OWL reasoner over the generated OWL2 KB and use the inferences to build a graph where OWL2 classes are nodes and subsumption and object restrictions are the edges. When generating a frame from a seed sense, we consider any sense that can reach it strictly through forward links to evoke that frame and add them to the frame's evoker set. Although we use the expressively constrained OWL2 for inference, the frames that we generate can be used in less constrained logics and are not inherently tied to OWL2's semantics. A small subgraph of what was generated for kill%2:35:00 is shown in Figure 2. At this point, we have generated frames and associated them with evokers, now we will turn our attention to frame elements. This is where the knowledge generated in (Allen et al., 2013) begins to differentiate our approach from shallower methods. Frame elements represent general types of participants that are typically involved in a frame as well as the role they play in that frame. Our frame elements describe what is necessary and sufficient for something to be classified as that element - this includes information about semantic type and the thematic relation that element has to the frame.

Observe that the paths between a frame and its evokers show how they evoke the frame. But most of these paths do not represent typical participants; there are a few cases like eliminator-%1:17:00 ("an agent that eliminates(kill%2:35:00) something") but they are comparatively rare. Instead, the evokers are better thought of as examples of one of the frame's elements. We propose using these examples to automatically derive frame elements and their semantic types. There are several possible ways to abstract and learn from such examples and one avenue we are exploring appears at the end of the next section.

4. Experiments

In this section we will present two different experiments, one evaluates the generation of evokers and also tests how well the thematic roles linking frame and evoker indicate their highlevel role in the frame. The second experiment was our first attempt at using those relations to learn a frame's elements.

4.1 Evaluating Evokers

First, we wanted to see if the evokers that the system generated were reasonable to a human. Many other systems, like those presented in section 2 , evaluate their results by comparing their generated evokers to the set of LU's in one of FrameNet's frames; but we feel FrameNet is inadequate for our purposes. As stated in the introduction, FrameNet has a coverage issue, so to use it as a standard will not provide a complete evaluation of our results. Although our goal is not to recreate FrameNet, we used existing frames to indicate what verbs would make good examples. We chose a subset of FrameNet frames discussed in (O'Connor, 2012) as well as those we thought would be interesting in terms of granularity (general or specific) and the types of evokers involved (concrete or abstract). We then chose one WordNet sense as a seed that we thought best represented the FrameNet based on their glosses.The evaluators were made up of 4 members from within our NLP group - not all of which are directly associated with this paper. For a select group of frames we generated 40 senses. A positive set of 20 evokers were chosen randomly from all of the generated evokers with glosses that did not explicitly mention the seed senses' lemma. This choice was made so that the evaluators had to actually consider the meaning of the senses instead of looking at surface features. We also chose a negative set of 20 random senses that were not in the evoker set but are WordNet sister terms (share the same direct hyponym) of one of the evokers. We pick the negative set this way because randomly choosing WordNet senses tends to generate obviously unrelated terms which would not help to understand what our system misses.

For each sense, we created a question consisting of its lemma, part of speech and gloss; we provide the same information for the seed sense as well. The evaluators were then given a multiple choice question that asks how the word was related to the seed sense based on the given glosses and their own knowledge. Each choice is meant to classify the relation between the two senses into one of the following relations: type of, causes, affected by, instrumental, preceded by, other, or does not evoke. The choices from one question in the evaluation of kill%2:35:00 are below.

- hit_squad is always a type/kind/subclass of kill
- a(n) hit_squad typically causes or intends to cause a(n) kill event
- a(n) hit_squad is typically directly changed by or experiences a(n) kill event
- a(n) hit_squad is typically intended to be used for a(n) kill event
- a(n) hit_squad is typically preceded by a(n) kill event
- a(n) hit_squad is typically related to a(n) kill event in a way not specified above
- a(n) hit_squad is not typically related to a(n) kill event in any way

For each survey we created two sets of results, the first combines all except the last choice, "does not evoke", into a super category named "evokes". Though this is a very shallow evaluation it is meant to assess how well we generate evokers - i.e. one of the task in creating a frame based resource.

Table 1. Results for a sample of generated frames.

Frame Seed	Category	Only Positive(20)	Positive and Negative(40)	Generated
kill%2:35:00	Binary	17/19 = .895	34/39 = .872	186
	All	6/14 = .43	23/33 = .7	
injure%2:29:00	Binary	16/18 = .800	27/38 = .711	109
	All	8/13 = .615	19/29 = .656	
ingest%2:34:00	Binary	15/19 = .789	29/38 = .763	1228
	All	3/14 = .214	17/32 = .531	
inform%2:32:00	Binary	9/13=.692	15/31 = .484	59
	All	0/9 = 0	6/23 = .261	
start%2:30:00	Binary	0/18 = 0	20/38 = .526	3720
	All	0/18 = 0	20/38 = .526	

The second set treats each of the seven choices individually; this evaluation is meant to determine if the thematic roles that linked an evoker to a seed sense are trustworthy enough to base our frame element generation on them. The first set of system responses is produced based on whether the word was in the evoker set, if it is then the system responds, "evokes"; otherwise it responds, "does not evoke". In the second evaluation we generate the system's response based a simple set of rules regarding the thematic role paths between the positive set of evokers and the seed term in the graph generated from the OWL2 KB. These rules are reproduced below:

- $\mathbf{y} \sqsubseteq \exists agent^{-}.\mathbf{x} \sqcup \exists cause^{-}.\mathbf{x} \sqcup \exists effect^{-}.\mathbf{x} \Rightarrow$ "cause" - y directly causes x
- $\mathbf{y} \sqsubseteq \exists agent.\mathbf{x} \sqcup \exists cause.\mathbf{x} \sqcup \exists effect.\mathbf{x} \Rightarrow$ "preceded by" - x directly causes y
- $\mathbf{y} \sqsubseteq \exists affected^{-}.\mathbf{x} \Rightarrow$ "affected by" - y is affected by some x
- $\mathbf{y} \sqsubseteq \exists instrument^{-}.\mathbf{x} \Rightarrow$ "instrument" - y is the instrument of
- $\mathbf{y} \sqsubseteq \mathbf{x} \Rightarrow$ "type of" - y is subsumed by x
- $\mathbf{y} \sqsubseteq \mathbf{ont{:}act} \sqcap \exists of.\mathbf{x} \Rightarrow ''typeof''$ - y is the act of x
- $\mathbf{y} \sqsubseteq \exists of^{-}.(\ \mathbf{by\%4{:}02{:}00} \sqcap \exists val.\mathbf{x}) \Rightarrow$ "type of" - y is accomplished by doing x
- otherwise the system answers "other"

For both cases we consider only those answers where more than half of the evaluators agreed. The end goal is to produce the same quality of knowledge as a human annotator; it tells us very little when there is not strong agreement amongst annotators. We then compute the agreement between the system and the majority answer. The table reports these numbers as <amount of correct answers for question subset>/<number of questions where human agreement is greater than half> = <ratio>. We also show results only considering the questions generated from the positive set to demonstrate the system's precision.

We can see from Table 1 that kill%2:35:00 and injure%2:29:00 performed the best overall. The evaluators agreed with kill%2:35:00's evoker set (binary) 17 times out of the 19 instances there was a majority consensus. We can see that overall inter-rater agreement was very high - there was only one instance of a split decisions between "evokes" and "does not evoke". Of course the inter-rater agreement and the agreement between the system and the majority answer are lower when considering the categories individually instead of as two distinct sets. The binary choice scores for all but start%2:30:00 are very encouraging. Considering the simplicity of our search method theses scores verify the suitability of dictionaries as input for the evoker generation task. Even when considering each category independently the scores are encouraging. Despite using a very simple set of classification rules the percentage of correct positives was significantly better than randomly selecting 1 of the 6 possible categories (.17). These scores indicate that the semantic paths between evoker and seed can be trusted enough to use to infer a frame's elements from them.

Our system performs noticeably worse when dealing with abstract concepts. There are two reasons for these instances of poor performance. Firstly, the semantic parser may mistag a word sense which could either include evokers that should not be there or exclude evokers that should. Mistaggings are expected to occur more often for abstract senses like those for the word, "get", or, "have", which generally have more senses with vague distinctions between their uses. Although this does happen for more specific senses, for example, the gloss for antiknock%3:00:00:leaded:00 is "suppressing or eliminating engine knock in combustion engines" and eliminating is tagged as eliminate%2:30:00 which means, "[to] kill in large numbers". When looking for evokers of kill%2:35:00 our system not only finds antiknock%3:00:00:leaded:00 but also octane_rating%1:23:00, "a measure of the antiknock properties of gasoline". Addressing this issues is complicated, obviously improving the TRIPS parser's tagging performance on WordNet glosses would help. This is actively being worked on; however, it is unlikely that the problem will be completely eliminated. Another area to explore is assessing the trustworthiness of the paths between a frame and its evokers. A trend we observed when exploring the erroneous evokers was that the paths between them and the frame tended to be long, had concepts relatively close to the top of the ontology, and involve underspecified roles like suchthat, parenthetical, or associatedwith. Taking into consideration the path's length and the roles and concepts involved we may be able to identify suspect evokers.

A second issue with abstract frames is that evaluators can overwhelmingly disagree with the system even though its reasoning is sound. For instance, the system says that batwing%1:06:00, "one of a pair of swinging doors (as at the entrance to a western saloon)". Evokes ingest%2:34:00 but no one agreed with that. The line of reasoning is this: batwings are located at bars, bars serve drinks, and drinks are suitable to ingest. The discrepancy between the evaluators and the correct line of reasoning indicates that our definition of what it means to evoke a sense is flawed and should be restricted in some way. A possible solution for this issue is very similar to the trust scheme presented above. We would take into account path length, the roles used, and the abstractness of the concepts involved. The success of this solution depends on our ability to use those factors to identify the threshold between human evaluators determining if a sense evokes a frame or not.

4.2 Generating Frames

Here we describe initial investigations into generating frames. Frame elements can be thought of as thematic roles in context. For instance, in FrameNet's Killing frame, an agent in the context of a kill action is called the Killer; in a more abstract frame, Commerce_goods-transfer, an entity is considered a buyer if it is an agent in the context of a buying event or if it is a recipient in the context of a selling event. With this in mind we base our element search on the thematic roles associated with the seed sense.

We begin by creating a set of examples of all of the target frame's elements. The example set is made up of not just evokers - where a word is defined in terms of its relation to the seed sense

- but also relations that define either the seed sense or a sense that is subsumed by the seed. What this means is that we use both murderee%1:18:00 (an evoker defined as a patient in a kill%2:35:00 event) and animal%1:03:00 which is not defined in terms of kill%2:35:00 but is used by one of kill%2:35:00's sub-senses, butcher%2:35:00, which is defined as a kill%2:35:00 event with an animal%1:03:00 as the patient. We then group the examples by the chain of thematic roles between it and the seed sense. If we were looking at a frame for kill%2:35:00 then concepts like poisoner%1:18:00 and bounty_hunter%1:18:00 would be grouped together because they both relate to kill%2:35:00 via the agent role. We first look for relationships that indicate special frame-frame relations. For instance, subsumption relations indicates a sub-frame relationship and a cause or effect relation (but not their inverse roles) indicates the frame-frame relations of the same name. For each group not handled by special relations, we run a hierarchical clustering algorithm that works as follows. We begin with a set of concepts grouped by their thematic roles as described above. For each pair of elements in the set, we find their anonymous least common subsumer (ALCS) based on a graph generated much the same way as the graph in Figure 2. The pair with the ALCS furtherest from the top of the ontology - i.e. the least general ALCS - is chosen to cluster together. From the set of concepts, the elements in the pair are removed and the ALCS of that pair is added. This is continued until the there is only one element in the set which subsumes every element in the original set.

After the hierarchy is produced we are left with a question: which of these clusters should form the frame's element for their associated thematic role. The root cluster is sometimes unsatisfyingly general. But, the leaf nodes and often the cluster directly above them are needlessly specific and end up just presenting knowledge generated in (Allen et al., 2013) in a different way. For this experiment, the cluster is searched from the top of the hierarchy down until a cluster no longer subsumes any of the elements in a set of seven very abstract concepts that convey little knowledge in the context of the roles in which they appear. Some examples include, physical-object, situation, and the root of the TRIPS ontology. Failing to find a specific enough cluster, the root of the cluster hierarchy is used.

Putting it all together we generated several frames from a small subset of seed senses that evoke kill%2:35:00. For an initial test, we limited the paths used to generate frame elements to only those that relate to the frame's seed sense via a path who's edge to the seed is an inverse role. e.g. **poisoner%1:18:00** – *agent*$^{-}$ $\rightarrow$ **kill%2:35:00**; but not paths like **mortal_enemy%1:18:00** – *pivot* $\rightarrow$ **want** – *theme* $\rightarrow$ **kill%2:35:00**. This choice was made because elements are meant to describe common participants in the frame and not the roles the frame itself participates in. The elements for kill%2:35:00 appears in Figure 3 presented in a fashion similar to FrameNet. Evaluating element generation this early in our experiments is difficult to do in a quantifiable way. Instead we will informally look at the frame elements generated for kill%2:35:00 and compare them to the core elements in FrameNet's Killing frame. Unlike FrameNet's LU's and frames, the frame elements do not suffer any serious coverage problems. Comparing the frame generated for kill%2:35:00 to the hand annotated Killing frame in FrameNet we find some analogues between the frame elements. For instance, both have agent (Killer), cause, and patient (victim) elements. Agent and patient have types that are comparatively more specific but not far off from their FrameNet analouges, (Killer and Victim). Theme is erroneously included here due to a parsing error in hit_list%1:10:00's gloss,

agent [] **Semantic Type:** person	person ⊓ (∀ _agent⁻.kill%2:35:00) Element Evokers: cutthroat%1:18:00, uxoricide%1:18:00, coup_de_grace%1:04:00, infanticide%1:18:00, death_squad%1:14:00, assassinator%1:18:00, matricide%1:18:00, patricide%1:18:00, parricide%1:18:00, poisoner%1:18:00, throttler%1:18:00, filicide%1:18:00, obliterator%1:17:00, eliminator%1:17:00, fratricide%1:18:00
cause []	∀ _cause⁻.kill%2:35:00 Element Evokers: bounty_hunter%1:18:00, suicide%1:18:00
effect []	∀ _effect⁻.kill%2:35:00 Element Evokers: body_count%1:23:00,
patient [] **Semantic Type:** animal	animal ⊓ (∀ _patient⁻.kill%2:35:00) Element Evokers:geryon%1:18:00, murderee%1:18:00, flypaper%1:27:00, blood_feud%1:04:00, scaffold%1:06:01
theme [] **Semantic Type:** referential-sem	referential-sem ⊓ (∀ _theme⁻.kill%2:35:00) Element Evokers: hit_list%1:10:00

Figure 3. The elements, their semantic types, as well as the evokers that are classified as that element.

"a list of victims to be eliminated(kill%2:35:00)". Effect also appears due to a parsing error, where body_count%1:23:00, "of troops killed in an operation or time period", is erroneously defined as the effect of some kill%2:35:00 event.

The frame is missing important elements as well, for instance, our system fails to generate analogues of FrameNet's Means and Instrument elements. Manually exploring the thematic role graph, we found several paths that would indicate a relation similar to Instrument, like this path between pesticide and kill[1]: **pesticide%1:27:00** – *theme*$^-$ → **use** – *of* → **purpose** – *val* → **kill%2:35:00** (pesticide is used for the purpose of killing). Poleaxe%1:06:01 has the same path to kill%2:35:00. However, those paths would be grouped in a different frame element than something like **halter%1:06:02** – *theme*$^-$ → **use** – *agent* → **hangman%1:18:00** – *agent*$^-$ → **kill%2:35:00** (a halter is used by a hangman to kill)[2]. One solution is to define abstract path patterns such as $\mathbf{x} - \mathit{theme}^- \rightarrow \mathbf{use} - ([\mathit{of} \rightarrow \mathbf{purpose} - \mathit{val}] | [\mathit{agent} \rightarrow \mathbf{person} - \mathit{agent}^-]) \rightarrow \mathbf{y} \Rightarrow \mathbf{x} - \_\mathit{instrument}^- \rightarrow \mathbf{y}$. Whether a small set of manually created patterns would suffice or if a large set of learned patters is needed remains to be seen.

5. Future Work

In addition to the fixes proposed in the previous section, we would like to explore how to combine several frames into a single, abstract frame. Specifically, we would like to discover transformation rules from one frame element to another. e.g. the patient of a kill event is also the patient of the resultant die event. This is imperative to eventually be able to generate abstract frames. Like Commerce_goods-transfer where we want to generate an element that is defined as being an agent in a buying event or a recipient in a selling event. The ability to abstract different points of view, like buying and selling, into a single frame would go a long way towards offering knowledge on par with FrameNet.

1. Using only paths ending with inverse roles prevents us from generating the corresponding element.
2. In our input hangman%1:18:00 is incorrectly related to kill via the *suchthat* relation rather than *agent*$^-$.

6. Conclusion

In this paper we presented work toward generating frame-like knowledge from WordNet. Our approach differs from the reviewed work in section 2 due to its use inference, high-level techniques, input, and work towards learning frame elements. It explores relatively deep knowledge generated in (Allen et al., 2013) to identify frames and their evokers and then performs semantically informed inference to discover frame elements. To assess our input we carried out a small survey designed to test if the relationship between a frame and its evokers fit human intuition. We then presented one method to learn frame elements from the processed input and compared it to FrameNet. Our ultimate goal is to use dictionary type resources to automatically generate a lexical KB of frames with greater coverage of evokers and granularity of frames than FrameNet. The purpose of the experiments presented above was to test our basic notions about how information in these sources relate to frames, evokers, and frame elements. Both experiments identified problem areas but the results are encouraging enough to warrant continued work.

Acknowledgements

The work presented in this paper was supported in part by the National Science Foundation (grant IIS-1012205) and The Office of Naval Research (grant N000141110417).

References

Allen, J., de Beaumont, W., Galescu, L., Orfan, J., Swift, M., & Teng, C. M. (2013). Automatically Deriving Event Ontologies for a Commonsense Knowledge Base. *Proceedings of the International Conference for Computational Semantics*.

Baker, C. F. (2012). FrameNet, current collaborations and future goals. *Language Resources and Evaluation*, *46*, 269–286.

Burchardt, A., Pennacchiotti, M., Thater, S., & Pinkal, M. (2009). Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering*, *15*, 527–550.

Johnson, C. R., Petruck, M. R. L., Baker, C. F., Charles, M., Ruppenhofer, E. J., & Fillmore, C. J. (2003). *FrameNet : Theory and Practice*.

Laparra, E., & Rigau, G. (2009). Integrating wordnet and framenet using a knowledge-based word sense disambiguation algorithm. *Proceedings of RANLP*, *9*.

Materna, J. (2012). LDA-Frames : An Unsupervised Approach. *Computational Linguistics and Intelligent Text Processing*, 376–387.

Miller, G. a. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*, 39–41.

O'Connor, B. (2012). Learning Frames from Text with an Unsupervised Latent Variable Model.

Shi, L., & Mihalcea, R. (2005). Putting Pieces Together : Combining FrameNet , VerbNet and WordNet. In *Computational linguistics and intelligent text processing*, 100–111.