The Holy Grail of Sense Definition: Creating a Sense-Disambiguated Corpus from Scratch

Anna Rumshisky
Dept. of Computer Science
Brandeis University
Waltham, MA
arum@cs.brandeis.edu

Marc Verhagen
Dept. of Computer Science
Brandeis University
Waltham, MA
marc@cs.brandeis.edu

Jessica L. Moszkowicz
Dept. of Computer Science
Brandeis University
Waltham, MA
jlittman@cs.brandeis.edu

Abstract

This paper presents a methodology for creating a gold standard for sense definition using Amazon's Mechanical Turk service. We demonstrate how this method can be used to create in a single step, quickly and cheaply, a lexicon of sense inventories and the corresponding sense-annotated lexical sample. We show the results obtained by this method for a sample verb and discuss how it can be improved to produce an exhaustive lexical resource. We then describe how such a resource can be used to further other semantic annotation efforts, using as an example the Generative Lexicon Mark-up Language (GLML) effort.

1 Introduction

The problem of defining a robust procedure for sense definition has been the holy grail of both lexicographic work and the sense annotation work done within the computational community. In recent years, there has been a number of initiatives to create gold standards for sense-annotated data to be used for the training and testing of sense disambiguation and induction algorithms. Such efforts have often been impeded by difficulties in selecting or producing a satisfactory sense inventory. Methodologically, defining a set of standards for creating sense inventories has been an elusive task.

In the past year, Mechanical Turk, introduced by Amazon as "artificial artificial intelligence", has been used successfully to create annotated data for a number of tasks, including sense disambiguation (Snow et al., 2008) as well as for creating a robust evaluation of machine translation systems by reading comprehension (Callison-Burch, 2009). Typically, complex annotation is split into simpler steps. Each step is farmed out to the non-expert annotators employed via Mechanical Turk (henceforce, MTurkers) in a form of a HIT (Human Intelligence Task), defined as a task that is hard to perform automatically, yet very easy to do for people.

In this paper, we propose a method using Mechanical Turk to create sense inventories from scratch and the corresponding sense-annotated lexical sample for any polysemous word. As with the NLP tasks for which MTurk has been used previously, this annotation is quite inexpensive and can be done very quickly. We test this method on a polysemous verb of medium difficulty and compare the results to the groupings created by a professional lexicographer. We then describe how this method can be used to create sense groupings for other words in order to create a sense enumerated lexicon. Finally, we describe one application in which such a resource would be invaluable, namely, the preprocessing step for the ongoing GLML annotation effort.

2 Problem of Sense Definition

In the past few years, a number of initiatives have been undertaken to create a standardized framework for the testing of word sense disambiguation (WSD) and word sense induction (WSI) systems, including the recent series of SENSEVAL competitions (Agirre et al., 2007; Mihalcea and Edmonds, 2004; Preiss and Yarowsky, 2001), and the shared semantic role labeling tasks at the CoNLL conference (Carreras and Marquez, 2005; Carreras and Marquez, 2004).

Training and testing such systems typically involves using a gold standard corpus in which each occurrence of the target word is marked with the appropriate sense from a given sense inventory. A number of such sense-tagged corpora have been developed over the past few years. Within the context of work done in the computational community, sense inventories used in such anntoation are usually taken out of machine-readable

```
PATTERN 1:  [[Anything]] crush [[Physical Object = Hard | Stuff = Hard]]
Explanation:
[[Anything]] damages or destroys [[Physical Object | Stuff = Hard]] by sudden and unexpected force

PATTERN 2:  [[Physical Object]] crush [[Human]]
Explanation:
[[Physical Object]] kills or injures [[Human]] by the pressure of irresistible force

PATTERN 3:  [[Institution | Human = Political or Miltary Leader]] crush [[Activity = Revolt | Independent
Action]]
Explanation:
[[Institution | Human = Political or Military Leader]] uses force to bring to an end [[Activity = Revolt |
Independent Action]] by [[Human Group]]

PATTERN 4:  [[Institution | Human 1 = Military or Political Leader ]] crush [[Human Group | Human 2 = LEXSET
]]
Explanation:
[[Institution | Human 1 = Military or Political Leader]] destroys or brings to an end the resistance of
[[Human Group | Human 2 = Military or Political Leader]]

PATTERN 5:  [[Event]] crush [[Human | Emotion]]
Explanation:
[[Event]] causes [[Human]] to lose hope or other positive [[Emotion]] and to feel bad
```

Figure 1: CPA entry for **crush**

dictionaries or lexical databases, such as WordNet (Fellbaum, 1998), Roget's thesaurus (Roget, 1962), Longman Dictionary of Contemporary English (LDOCE, 1978), Hector project, etc. In some cases inventories are (partially or fully) constructed or adapted from an existing resource in a pre-annotation stage, as in PropBank (Palmer et al., 2005) or OntoNotes (Hovy et al., 2006). The quality of the annotated corpora depends directly on the selected sense inventory, so, for example, SemCor (Landes et al., 1998), which uses WordNet synsets, inherits all the associated problems, including using senses that are too fine-grained and in many cases poorly distinguished. At the recent Senseval competitions (Mihalcea et al., 2004; Snyder and Palmer, 2004; Preiss and Yarowsky, 2001), the choice of a sense inventory also frequently presented problems, spurring the efforts to create coarser-grained sense inventories (Navigli, 2006; Hovy et al., 2006; Palmer et al., 2007; Snow et al., 2007). Inventories derived from WordNet by using small-scale corpus analysis and by automatic mapping to top entries in Oxford Dictionary of English were used in the most recent workshop on semantic evaluation, Semeval-2007 (Agirre et al., 2007).

Establishing a set of senses available to a particular lexical item is a task that is notoriously difficult to formalize. This is especially true for polysemous verbs with their constellations of related meanings. In lexicography, "lumping and splitting" senses during dictionary construction – i.e. deciding when to describe a set of usages as a separate sense – is a well-known problem (Hanks and Pustejovsky, 2005; Kilgarriff, 1997; Apresjan, 1973). It is often resolved on an ad-hoc basis, resulting in numerous cases of "overlapping senses", i.e. instances when the same occurrence may fall under more than one sense category simultaneously. Within lexical semantics, there has also been little consent on theoretical criteria for sense definition, while a lot of work has been devoted to questions such as when the context selects a distinct sense and when it merely modulates the meaning, what is the regular relationship between related senses, what compositional processes are involved in sense selection, and so on (Pustejovsky, 1995; Cruse, 1995; Apresjan, 1973).

Several current resource-oriented projects attempt to formalize the procedure of creating a sense inventory. FrameNet (Ruppenhofer et al., 2006) attempts to organize lexical information in terms of script-like semantic frames, with semantic and syntactic combinatorial possibilities specified for each frame-evoking lexical unit (word/sense pairing). Corpus Pattern Analysis (CPA) (Hanks and Pustejovsky, 2005) attempts to catalog norms of usage for individual words, specifying them in terms of context patterns.

A number of other projects use somewhat similar corpus analysis techniques. In PropBank (Palmer et al., 2005), verb senses were defined based on their use in Wall Street Journal corpus and specified in terms of framesets which consist of a set of semantic roles for the arguments of a particular sense. In the OntoNotes project (Hovy et al., 2006), annotators use small-scale corpus analysis to create sense inventories derived by grouping together WordNet senses. The procedure is restricted to maintain 90% inter-annotator agreement.

Annotating each instance with a sense, and es-

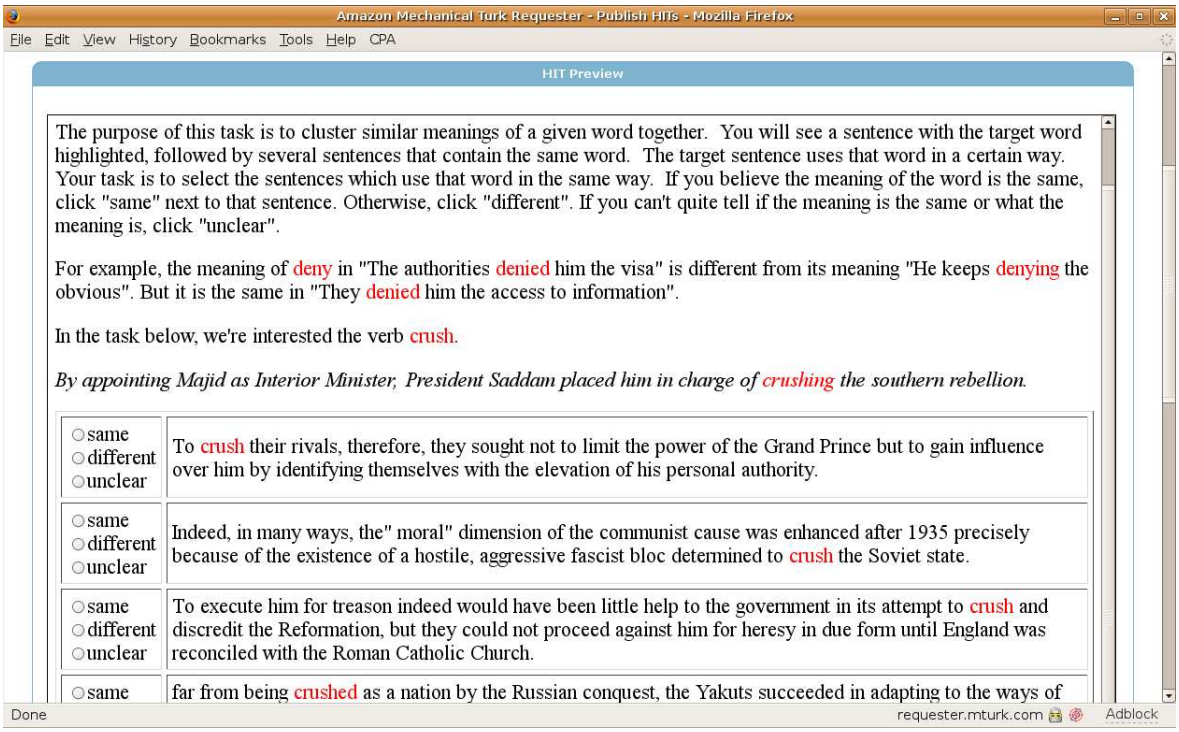

Figure 2: Task interface and instructions for the HIT presented to the non-expert annotators

pecially the creation of a sense inventory is typically very labor-intensive and requires expert annotation. We propose an empirical solution to the problem of sense definition, building both a sense inventory and an annotated corpus at the same time, using minimal expert contribution. The task of sense definition is accomplished empirically using native speaker, but non-expert, annotators. In the next section, we report on an experiment using this procedure for sense definition, and evaluate the quality of the obtained results.

3 Solution to the Problem (Experiment)

3.1 Task Design

We offered MTurkers an annotation task designed to imitate the process of creating clusters of examples used in CPA (CPA, 2009). In CPA, the lexicographer sorts the set of instances for the given target word into groups using an application that allows him or her to mark each instance in a concordance with a sense number. For each group of examples, the lexicographer then records a pattern that captures the relevant semantic and syntactic context features that allow us to identify the corresponding sense of the target word (Hanks and Pustejovsky, 2005; Pustejovsky et al., 2004; Rumshisky et al., 2006).

For this experiment, we used the verb *crush*, which has 5 different sense-defining patterns assigned to it in the CPA verb lexicon, and in which some senses appear to be metaphorical extensions of the primary physical sense. We therefore view it as a verb of medium difficulty both for sense inventory creation and for annotation. The patterns from the CPA verb lexicon for *crush* are given in Figure 1. The CPA verb lexicon has 350 sentences from the BNC that contain the verb *crush*, sorted according to these patterns.

The task was designed as a sequence of annotation rounds, with each round creating a cluster corresponding to one sense. MTurkers were first given a set of sentences containing the target verb, and one sentence that is randomly selected from this set as a prototype sentence. They were then asked to identify, for each sentence, whether the target verb is used in the same way as in the prototype sentence. If the sense was unclear or it was impossible to tell, they were instructed to pick the “unclear” option. We took the example sentences from the CPA verb lexicon for *crush*. Figure 2 shows the first screen displayed to MTurkers for this HIT. Ten examples were presented on each screen. Each example was annotated by 5 MTurkers.

After the first round of annotation was complete, the sentences that were judged as similar

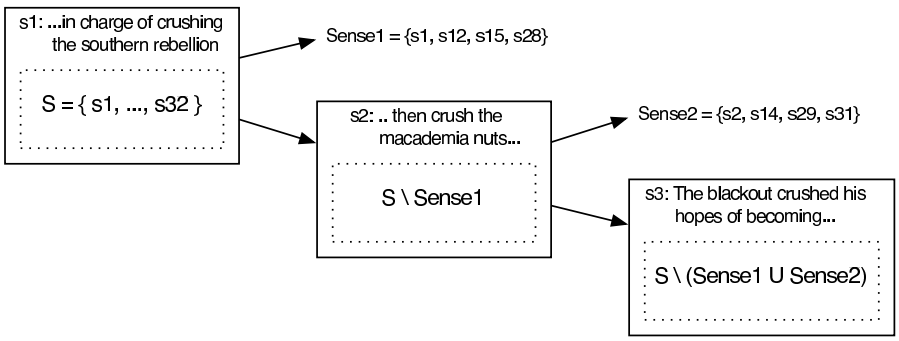

Figure 3: Sense-defining iterations on a set of sentences S.

to the prototype sentence by the majority vote (3 or more out of 5 annotators) were set apart into a separate cluster corresponding to one sense, and excluded from the set used in further rounds. The procedure was repeated with the remaining set, i.e. a new prototype sentence was selected, and the remaining examples were presented to the annotators. See Figure 3 for an illustration.

This procedure was repeated until all the remaining examples were classified as "unclear" by the majority vote, or no examples remained. Since some misclassifications are bound to occur, we stopped the process when the remaining set contained 7 examples, several of which were unclear, and others judged to be misclassification by an expert.

Each round took approximately 30 minutes to an hour to complete, depending on the number of sentences in that round. Each set of 10 sentences took approximately 1 minute on the average to complete, and the annotator received $0.03 USD as compensation. The total sum spent on this experiment did not exceed $10 USD.

3.2 Evaluation

3.2.1 Comparison against expert judgements

We evaluated the results of the experiment against the gold standard created by a professional lexicographer for the CPA verb lexicon. In the discussion below, we will use the term *cluster* to refer to the clusters created by non-experts as described above. Following the standard terminology in sense induction tasks, we will refer to the clusters created by the lexicographer as *sense classes*.

We used two measures to evaluate how well the clusters produced by non-experts matched the classes in the CPA verb lexicon: the *F-score* and *Entropy*. The *F-score* (Zhao et al., 2005; Agirre and Soroa, 2007) is a set-matching measure. Precision, recall, and their harmonic mean (van Rijsbergen's F-measure) are computed for each cluster/sense class pair. Each cluster is then matched to the class with which it achieves the highest F-measure.[1] The *F-score* is computed as a weighted average of the F-measure values obtained for each cluster.

Entropy-related measures, on the other hand, evaluate the overall quality of a clustering solution with respect to the gold standard set of classes. *Entropy* of a clustering solution, as it has been used in the literature, evaluates how the sense classes are distributed with each derived cluster. It is computed as a weighted average of the entropy of the distribution of senses within each cluster:

$$\textbf{Entropy}(C,S) = -\sum_i \frac{|c_i|}{n} \sum_j \frac{|c_i \cap s_j|}{|c_i|} \log \frac{|c_i \cap s_j|}{|c_i|} \quad (1)$$

where $c_i \in C$ is a cluster from the clustering solution C, and $s_j \in S$ is a sense from the sense assignment S.

We use the standard entropy definition (Cover and Thomas, 1991), so, unlike in the definition used in some of the literature (Zhao et al., 2005), the terms are not multiplied by the inverse of the log of the number of senses. The values obtained for the two measures are shown in the "initial" column of Table 1.

	initial	merged
F-score	65.8	93.0
Entropy	1.1	0.3

Table 1: F-score and entropy of non-expert clustering against expert classes. Second column shows evaluation against merged expert classes.

While these results may appear somewhat disappointing, it is important to recall that the CPA

[1] Multiple clusters may therefore map to the same class.

verb lexicon specifies classes corresponding to syntactic and semantic patterns, whereas the non-experts judgements were effectively producing clusters corresponding to senses. Therefore, these results may partially reflect the fact that several CPA context patterns may represent a single sense, with patterns varying in syntactic structure and/or the encoding of semantic roles relative to the described event.

We investigated this possibility, first automatically, and then through manual examination. Table 2 shows the overlap between non-expert clusters and expert classes. These numbers clearly suggest

		expert classes				
		1	2	3	4	5
non-expert clusters	1	0	2	120	65	9
	2	83	45	0	3	0
	3	0	0	2	1	10

Table 2: Overlap between non-expert clusters and expert classes.

that cluster 1 mapped into classes 3 and 4, cluster 2 mapped into classes 1 and 2, and cluster 3 mapped mostly into class 5. Indeed, here are the prototype sentences associated with each cluster:

C1 By appointing Majid as Interior Minister, President Saddam placed him in charge of **crushing** the southern rebellion.

C2 The lighter woods such as balsa can be **crushed** with the finger.

C3 This time the defeat of his hopes didn't **crush** him for more than a few days.

It is easy to see that the data above indeed reflects the appropriate mapping to the CPA verb lexicon patterns for *crush*, as shown in Figure 1.

The second column of Table 1 shows the values for *F-score* and *Entropy* computed for the case when the corresponding classes (i.e. classes 1 and 2 and classes 3 and 4) are merged.

3.2.2 Inter-annotator agreement

We computed pairwise agreement for all participating MTurkers. For each individual sentence, we looked at all pairs of judgements given by the five annotators, and considered the total number of agreements and disagreements in such pairs.

We computed Fleiss' kappa statistic for all the judgements collected from the group of participating MTurkers in the course of this experiment. The obtained value of kappa was $\kappa = 57.9$, with the actual agreement value 79.1%. The total number of instances judged was 516.

It is remarkable that these figures were obtained even though we performed no weeding of non-experts which performed poorly on the task. As suggested previously (Callison-Burch, 2009), it is fairly easy to implement the filtering out of MTurkers that do not perform well on the task. In particular, an annotator whose agreement with the other annotators is significantly below the average for the other annotators could be excluded from the majority voting described in Section 3.1.

The data in Table 3 should give some idea regarding the distribution of votes in majority voting.

No. of votes	% of judged instances
3 votes	12.8%
4 votes	29.8%
5 votes	55.2%

Table 3: The number of winning votes in majority voting.

4 Sense-Annotated Lexicon

The methodology we used in the described experiment has potential for producing quickly and efficiently, "from the ground up", both the sense inventories and the annotated data for hundreds of polysemous words. Task sets for multiple polysemous words can be in parallel, potentially reducing the overall time for such a seemingly monstrous effort to mere weeks.

However, a large-scale effort would require some serious planning and a more sophisticated workflow than the one used for this experiment. Mechanical Turk provides developer tools and an API that can help automate this process further. In particular, the following steps need to be automated: (i) construction of HITs in subsequent iterations, and (ii) management of the set of MTurkers (when to accept their results, what requirements to specify, and so forth). In the present experiment, we performed no quality checks on the ratings of each individual MTurker. Also, the first step was only partially automated, that is, after each set of HITs was completed, human intervention was required to run a set of scripts that produce and set up the next set of HITs using the remaining sentences. In addition, some

more conceptual issues need to be addressed:

The clarity of the sense distinctions. High inter-annotator agreement in the present experiment seems to suggest that *crush* has easily identifiable senses. It is possible, and even likely, that creating a consistent sense inventory would be much harder for other polysemous verbs, many of which are notorious for having convoluted constellations of inter-related and overlapping senses (see, for example, the discussion of *drive* in Rumshisky and Batiukova (2008)).

The optimal number of MTurkers. We chose to use five MTurkers per HIT based on observations in Snow et al. (2008), but we have no data supporting that five is optimal for this task. This relates to the previous issue since the optimal number can vary given the complexity of the task.

The quality of prototype sentences. We selected the prototype sentences completely randomly in the present experiment. However, it is obvious that if the sense of the target word is somewhat unclear in the prototype sententence, the quality of the associated cluster should fall drastically. This problem could be remedied by introducing an additional step, where another set of MTurkers would be asked to judge the clarity of a particular examplar sentence. If the current prototype sentence is judged to be unclear by the majority of MTurkers, it would be removed from the data set, and another prototype sentence would be randomly selected and judged.

Finally, we need to contemplate the kind of resource that this approach generates. It seems clear that the resulting resource will contain more coarse-grained sense distinctions than those observed in CPA, as evidenced by the *crush* example. But how it will actually work out for a large lexicon is still an empirical question.

5 The GLML Annotation Effort

In this section, we discuss some of the implications of the above approach for an existing semantic annotation effort. More explicitly, we survey how sense clustering can underpin the annotation of type coercion and argument selection and provide an outline for how such an annotation effort could proceed.

5.1 Preliminaries

In Pustejovsky et al. (2009) and Pustejovsky and Rumshisky (2009), a procedure for annotating argument selection and coercion was laid out. The aim was to identify the nature of the compositional relation rather than merely annotating surface types of entities involved in argument selection. Consider the example below.

(a) *Mary* called yesterday.
(b) *The Boston office* called yesterday.

The distinction between (a) and (b) can be described by semantic typing of the agent, but not by sense tagging and role labeling as used by FrameNet (Ruppenhofer et al., 2006) and PropBank (Palmer et al., 2005). Pustejovsky et al. (2009) focus on two modes of composition: pure selection (the type a function requires is directly satisfied by the argument) and type coercion (the type a function requires is imposed on the argument by exploitation or introduction). They describe three tasks for annotating compositional mechanisms: verb-based annotation, qualia in modification structures, and type selection in modification of dot objects. All three tasks involve two stages: (i) a data preparation phase with selection of annotation domain and construction of sense inventories and (ii) the annotation phase.

For the purpose of this paper we focus on the first annotation task: choosing which selectional mechanism (pure selection or coercion) is used by the predicate over a particular argument. The data preparation phase for this task consists of:

(1) Selecting a set of highly coercive target verbs.
(2) Creating a sense inventory for each verb.
(3) Assigning type templates to each sense.

For example, the "refuse to grant" sense of *deny* in *The authorities denied him the visa* will be associated with the template [HUMAN deny HUMAN ENTITY]. The same sense of the verb in *They denied shelter to refugees* will be associated with the template [HUMAN deny ENTITY to HUMAN].

After this, the annotation phase proceeds in two steps. First, for each example, annotators select the sense of a verb. Then, the annotators specify whether, given the sense selected, the argument matches the type specified in the template. If the type does not match, we have an instance of type coercion and the annotator will be asked what the

type of the argument is. For this purpose, a list of about twenty types is provided.

Two issues pop up rather pressingly. First, how should the sense inventory be created? This particular choice strongly influences the kind of coercions that are possible and it would be wise to avoid theoretical bias as much as possible. Pustejovsky et al. (2009) chose to use a lexicographically oriented sense inventory provided by the CPA project, but the method described here suggests another route. While the inventory provided by CPA is certainly of very high quality, its use for an annotation effort like GLML is limited because the inventory itself is limited by the need for expert lexicographers to perform a very labor-intensive analysis.

The second problem involves the shallow type system. Notice that this 'type system' informs both the template creation and the selection of actual types of the argument. It is unclear how to pick these types and, again, the choice of types defines the kinds of coercions that are distinguished.

5.2 Adapting GLML Annotation

We now provide a few adaptations of the procedure outlined in the previous subsection. We believe that these changes provide a more bottom-up procedure in distinguishing pure selection and type coercion in argument selection.

In Section 3, we described a fast bottom-up procedure to create sense inventories. Consider that after this procedure we have the following artifacts: (i) a set of senses, represented as clusters of examples, and (ii) the resulting clusters of sentences that illustrate the usage of each sense.

We can now put the resulting sense inventory and the associated examples to work, seeing contributions to both the preparation and annotation phase. Most of the work occurs in the preparation phase, in defining the templates and the shallow list of types. Firstly, the clusters define the sense inventory called for in the data preparation phase of the argument selection task. The clusters also trivially provide for the sense selection, the first step of the annotation phase. Secondly, the exemplars can guide a trained linguist in creating the templates. For each sense, the linguist/lexicographer needs to distinguish between instances of pure selection and instances of type coercion. For each set, a succinct way to describe the type of the argument needs to be defined.

The second item above has a desirable implication for the annotation process. Note that the guidelines can focus on each target verb separately because the set of available types for selection and coercion are defined for each target verb individually. In addition, the set of exemplars provides illustrations for each type. In general, the semantic types can be considered shorthands for sets of easily available data that the annotator can use.

Overall, we have taken the specialist out of the construction of the sense inventory. However, it is still up to the expert to analyze and process the clusters defined by MTurkers. For each cluster of examples, one or more type templates need to be created by the expert. In addition, the expert needs to analyze the types involved in type coercion and add these types to the list of possible type coercions for the target verb.

6 Conclusion

In this paper, we have presented a method for creating sense inventories and the associated sense-tagged corpora from the ground up, without the help of expert annotators. Having a lexicon of sense inventories built in this way would alleviate some of the burden on the expert in many NLP tasks. For example, in CPA, splitting and lumping can be done by the non-expert annotators. The task of the expert is then just to formulate the syntactic and semantic patterns that are characteristic for each sense.

We discussed one application for the resource that can be produced by this method, namely, the GLML annotation effort, but the ability to quickly and inexpensively generate sense clusters without the involvement of experts would assist in a myriad of other projects that involve word sense disambiguation or induction.

References

E. Agirre and A. Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of SemEval-2007*, pages 7–12, Prague, Czech Republic, June. Association for Computational Linguistics.

E. Agirre, L. Màrquez, and R. Wicentowski, editors. 2007. *Proceedings of SemEval-2007*. Association for Computational Linguistics, Prague, Czech Republic, June.

Ju. Apresjan. 1973. Regular polysemy. *Linguistics*, 142(5):5–32.

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using amazon's mechanical turk. In *Proceedings of EMNLP 2009*.

X. Carreras and L. Marquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL-2004*, pages 89–97.

X. Carreras and L. Marquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of CoNLL-2005*.

T. Cover and J. Thomas. 1991. *Elements of Information Theory*. John Wiley & sons.

CPA. 2009. Corpus Pattern Analysis, CPA Project Website, *http://nlp.fi.muni.cz/projekty/cpa/*. Masaryk University, Brno, Czech Republic.

D. A. Cruse. 1995. Polysemy and related phenomena from a cognitive linguistic viewpoint. In Patrick St. Dizier and Evelyne Viegas, editors, *Computational Lexical Semantics*, pages 33–49. Cambridge University Press, Cambridge, England.

C. Fellbaum, editor. 1998. *Wordnet: an electronic lexical database*. MIT Press.

P. Hanks and J. Pustejovsky. 2005. A pattern dictionary for natural language processing. *Revue Française de Linguistique Appliquée*.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of HLT-NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA, June. Association for Computational Linguistics.

A. Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31:91–113.

S. Landes, C. Leacock, and R.I. Tengi. 1998. Building semantic concordances. In C. Fellbaum, editor, *Wordnet: an electronic lexical database*. MIT Press, Cambridge (Mass.).

LDOCE. 1978. Longman Dictionary of Contemporary English. Longman Group Ltd, Harlot, England.

R. Mihalcea and P. Edmonds, editors. 2004. *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, July. Association for Computational Linguistics.

R. Mihalcea, T. Chklovski, and A. Kilgarriff. 2004. The Senseval-3 English lexical sample task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3*, pages 25–28, Barcelona, Spain, July. Association for Computational Linguistics.

R. Navigli. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of COLING-ACL 2006*, pages 105–112, Sydney, Australia, July. Association for Computational Linguistics.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

M. Palmer, H. Dang, and C. Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Journal of Natural Language Engineering*.

J Preiss and D. Yarowsky, editors. 2001. *Proceedings of the Second Int. Workshop on Evaluating WSD Systems (Senseval 2)*. ACL2002/EACL2001.

J. Pustejovsky and A. Rumshisky. 2009. SemEval-2010 Task 7: Argument Selection and Coercion. *NAACL HLT 2009 Workshop on Semantic Evaluations: Recent Achievements and Future Directions*.

J. Pustejovsky, P. Hanks, and A. Rumshisky. 2004. Automated Induction of Sense in Context. In *COLING 2004, Geneva, Switzerland*, pages 924–931.

J. Pustejovsky, A. Rumshisky, J. Moszkowicz, and O. Batiukova. 2009. GLML: Annotating argument selection and coercion. *IWCS-8*.

J. Pustejovsky. 1995. *Generative Lexicon*. Cambridge (Mass.): MIT Press.

P. M. Roget, editor. 1962. *Roget's Thesaurus*. Thomas Cromwell, New York, 3 edition.

A. Rumshisky and O. Batiukova. 2008. Polysemy in verbs: systematic relations between senses and their effect on annotation. In *HJCL-2008*, Manchester, England. submitted.

A. Rumshisky, P. Hanks, C. Havasi, and J. Pustejovsky. 2006. Constructing a corpus-based ontology using model bias. In *FLAIRS 2006*, Melbourne Beach, Florida, USA.

J. Ruppenhofer, M. Ellsworth, M. Petruck, C. Johnson, and J. Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*.

R. Snow, S. Prakash, D. Jurafsky, and A. Ng. 2007. Learning to merge word senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1005–1014. Association for Computational Linguistics.

R. Snow, B. OConnor, D. Jurafsky, and A.Y. Ng. 2008. Cheap and fastbut is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP 2008*.

B. Snyder and M. Palmer. 2004. The english all-words task. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3*, pages 41–43, Barcelona, Spain, July. Association for Computational Linguistics.

Ying Zhao, George Karypis, and Usama M. Fayyad. 2005. Hierarchical clustering algorithms for document datasets. 10:141–168.