Active Learning for the Identification of Nonliteral Language *

Julia Birke and Anoop Sarkar
School of Computing Science, Simon Fraser University
Burnaby, BC, V5A 1S6, Canada
jbirke@alumni.sfu.ca, anoop@cs.sfu.ca

Abstract

In this paper we present an active learning approach used to create an annotated corpus of literal and nonliteral usages of verbs. The model uses nearly unsupervised word-sense disambiguation and clustering techniques. We report on experiments in which a human expert is asked to correct system predictions in different stages of learning: (i) after the last iteration when the clustering step has converged, or (ii) during each iteration of the clustering algorithm. The model obtains an f-score of 53.8% on a dataset in which literal/nonliteral usages of 25 verbs were annotated by human experts. In comparison, the same model augmented with active learning obtains 64.91%. We also measure the *number* of examples required when model confidence is used to select examples for human correction as compared to random selection. The results of this active learning system have been compiled into a freely available annotated corpus of literal/nonliteral usage of verbs in context.

1 Introduction

In this paper, we propose a largely automated method for creating an annotated corpus of literal vs. nonliteral usages of verbs. For example, given the verb "pour", we would expect our method to identify the sentence "Custom demands that cognac be *poured* from a freshly opened bottle" as literal, and the sentence "Salsa and rap music *pour* out of the windows" as nonliteral, which, indeed, it does.

*This research was partially supported by NSERC, Canada (RGPIN: 264905). We would like to thank Bill Dolan, Fred Popowich, Dan Fass, Katja Markert, Yudong Liu, and the anonymous reviewers for their comments.

We reduce the problem of nonliteral language recognition to one of word-sense disambiguation (WSD) by redefining *literal* and *nonliteral* as two different senses of the same word, and we adapt an existing similarity-based word-sense disambiguation method to the task of separating usages of verbs into literal and nonliteral clusters. Note that treating this task as similar to WSD only means that we use features from the local context around the verb to identify it as either literal or non-literal. It does not mean that we can use a classifier trained on WSD annotated corpora to solve this issue, or use any existing WSD classification technique that relies on supervised learning. We do not have any annotated data to train such a classifier, and indeed our work is focused on building such a dataset. Indeed our work aims to first discover reliable seed data and then bootstrap a literal/nonliteral identification model. Also, we cannot use any semi-supervised learning algorithm for WSD which relies on reliably annotated seed data since we do not possess any reliably labeled data (except for our test data set). However we do exploit a noisy source of seed data in a nearly unsupervised approach augmented with active learning. Noisy data containing example sentences of literal and nonliteral usage of verbs is used in our model to cluster a particular instance of a verb into one class or the other. This paper focuses on the use of active learning using this model. We suggest that this approach produces a large saving of effort compared to creating such an annotated corpus manually.

An active learning approach to machine learning is one in which the learner has the ability to influence the selection of at least a portion of its training data. In our approach, a clustering algorithm for literal/nonliteral recognition tries to annotate the examples that it can, while in each iteration it sends a small set of examples to a human expert to annotate, which in turn provides additional benefit to the bootstrapping process. Our active learn-

ing method is similar to the Uncertainty Sampling algorithm of (Lewis & Gale, 1994) but in our case interacts with iterative clustering. As we shall see, some of the crucial criticisms leveled against uncertainty sampling and in favor of Committee-based sampling (Engelson & Dagan, 1996) do not apply in our case, although the latter may still be more accurate in our task.

2 Literal vs. Nonliteral Identification

For the purposes of this paper we will take the simplified view that *literal* is anything that falls within accepted selectional restrictions ("he was forced to eat his spinach" vs. "he was forced to eat his words") or our knowledge of the world ("the sponge absorbed the water" vs. "the company absorbed the loss"). *Nonliteral* is then anything that is "not literal", including most tropes, such as metaphors, idioms, as well as phrasal verbs and other anomalous expressions that cannot really be seen as *literal*. We aim to automatically discover the contrast between the standard set of selectional restrictions for the literal usage of verbs and the non-standard set which we assume will identify the nonliteral usage.

Our identification model for literal vs. nonliteral usage of verbs is described in detail in a previous publication (Birke & Sarkar, 2006). Here we provide a brief description of the model so that the use of this model in our proposed active learning approach can be explained.

Since we are attempting to reduce the problem of literal/nonliteral recognition to one of word-sense disambiguation, we use an existing similarity-based word-sense disambiguation algorithm developed by (Karov & Edelman, 1998), henceforth KE. The KE algorithm is based on the principle of attraction: similarities are calculated between sentences containing the word we wish to disambiguate (the *target word*) and collections of seed sentences (*feedback sets*). It requires a *target set* – the set of sentences containing the verbs to be classified into literal or nonliteral – and the seed sets: the *literal feedback set* and the *nonliteral feedback set*. A target set sentence is considered to be attracted to the feedback set containing the sentence to which it shows the highest similarity. Two sentences are similar if they contain similar words and two words are similar if they are contained in similar sentences. The resulting *transitive similarity* allows us to defeat the *knowledge acquisition bottleneck* – i.e. the low likelihood of finding all possible usages of a word in a single corpus. Note that the KE algorithm concentrates on similarities in the way sentences use the target literal or nonliteral word, not on similarities in the meanings of the sentences themselves.

Algorithms 1 and 2 summarize our approach. Note that $p(w, s)$ is the unigram probability of word w in sentence s, normalized by the total number of words in s. We omit some details about the algorithm here which do not affect our discussion about active learning. These details are provided in a previous publication (Birke & Sarkar, 2006).

As explained before, our model requires a target set and two seed sets: the literal feedback set and the nonliteral feedback set. We do not explain the details of how these feedback sets were constructed in this paper, however, it is important to note that the feedback sets themselves are noisy and not carefully vetted by human experts. The literal feedback set was built from WSJ newswire text, and for the nonliteral feedback set, we use expressions from various datasets such as the Wayne Magnuson English Idioms Sayings & Slang and George Lakoff's Conceptual Metaphor List, as well as example sentences from these sources. These datasets provide lists of verbs that may be used in a nonliteral usage, but we cannot explicitly provide only those sentences that contain nonliteral use of that verb in the nonliteral feedback set. In particular, knowing that an expression *can* be used nonliterally does not mean that you can tell when it *is* being used nonliterally. In fact even the literal feedback set has noise from nonliteral uses of verbs in the news articles. To deal with this issue (Birke & Sarkar, 2006) provides automatic methods to clean up the feedback sets during the clustering algorithm. Note that the feedback sets are not cleaned up by human experts, however the test data is carefully annotated by human experts (details about inter-annotator agreement on the test set are provided below). The test set is not large enough to be split up into a training and test set that can support learning using a supervised learning method.

The sentences in the target set and feedback sets were augmented with some shallow syntactic information such as part of speech tags provided

Algorithm 1 *KE-train*: (Karov & Edelman, 1998) algorithm adapted to literal/nonliteral identification

Require: $\mathcal{S}$: the set of sentences containing the *target word* (each sentence is classified as literal/nonliteral)
Require: $\mathcal{L}$: the set of literal seed sentences
Require: $\mathcal{N}$: the set of nonliteral seed sentences
Require: $\mathcal{W}$: the set of words/features, $w \in s$ means w is in sentence s, $s \ni w$ means s contains w
Require: ϵ: threshold that determines the stopping condition

w-sim$_0(w_x, w_y) := 1$ *if* $w_x = w_y, 0$ *otherwise*
s-sim$^I_0(s_x, s_y) := 1$, for all $s_x, s_y \in \mathcal{S} \times \mathcal{S}$ where $s_x = s_y$, 0 otherwise
$i := 0$
while (**true**) **do**
  s-sim$^L_{i+1}(s_x, s_y) := \sum_{w_x \in s_x} p(w_x, s_x) \max_{w_y \in s_y}$ *w-sim*$_i(w_x, w_y)$, for all $s_x, s_y \in \mathcal{S} \times \mathcal{L}$
  s-sim$^N_{i+1}(s_x, s_y) := \sum_{w_x \in s_x} p(w_x, s_x) \max_{w_y \in s_y}$ *w-sim*$_i(w_x, w_y)$, for all $s_x, s_y \in \mathcal{S} \times \mathcal{N}$
  for $w_x, w_y \in \mathcal{W} \times \mathcal{W}$ **do**
    $$\textit{w-sim}_{i+1}(w_x, w_y) := \begin{cases} i = 0 & \sum_{s_x \ni w_x} p(w_x, s_x) \max_{s_y \ni w_y} \textit{s-sim}^I_i(s_x, s_y) \\ \text{else} & \sum_{s_x \ni w_x} p(w_x, s_x) \max_{s_y \ni w_y} \{\textit{s-sim}^L_i(s_x, s_y), \textit{s-sim}^N_i(s_x, s_y)\} \end{cases}$$
  end for
  if $\forall w_x, \max_{w_y}\{$*w-sim*$_{i+1}(w_x, w_y) -$ *w-sim*$_i(w_x, w_y)\} \leq \epsilon$ **then**
    break # algorithm converges in $\frac{1}{\epsilon}$ steps.
  end if
  $i := i + 1$
end while

by a statistical tagger (Ratnaparkhi, 1996) and SuperTags (Bangalore & Joshi, 1999).

This model was evaluated on 25 target verbs:

> absorb, assault, die, drag, drown, escape, examine, fill, fix, flow, grab, grasp, kick, knock, lend, miss, pass, rest, ride, roll, smooth, step, stick, strike, touch

The verbs were carefully chosen to have varying token frequencies (we do not simply learn on frequently occurring verbs). As a result, the target sets contain from 1 to 115 manually annotated sentences for each verb to enable us to measure accuracy. The annotations were not provided to the learning algorithm: they were only used to evaluate the test data performance. The first round of annotations was done by the first annotator. The second annotator was given no instructions besides a few examples of literal and nonliteral usage (not covering all target verbs). The authors of this paper were the annotators. Our inter-annotator agreement on the annotations used as test data in the experiments in this paper is quite high. κ (Cohen) *and* κ (S&C) on a random sample of 200 annotated examples annotated by two different annotators was found to be 0.77. As per ((Di Eugenio & Glass, 2004), cf. refs therein), the standard assessment for κ values is that tentative conclusions on agreement exists when $.67 \leq \kappa < .8$, and a definite conclusion on agreement exists when $\kappa \geq .8$.

In the case of a larger scale annotation effort, having the person leading the effort provide one or two examples of literal and nonliteral usages for each target verb to each annotator would almost certainly improve inter-annotator agreement.

The algorithms were evaluated based on how accurately they clustered the hand-annotated sentences. Sentences that were attracted to neither cluster or were equally attracted to both were put in the opposite set from their label, making a failure to cluster a sentence an incorrect clustering.

Evaluation results were recorded as *recall*, *precision*, and *f-score* values. *Literal recall* is defined as *(correct literals in literal cluster / total correct literals)*. *Literal precision* is defined as *(correct literals in literal cluster / size of literal cluster)*. If there are no literals, *literal recall* is 100%; *literal precision* is 100% if there are no nonliterals in the literal cluster and 0% otherwise. The *f-score* is defined as *(2 ·*

Algorithm 2 *KE-test*: classifying literal/nonliteral

For any sentence $s_x \in \mathcal{S}$
if $\max_{s_y}$ *s-sim*$^L(s_x, s_y)$ $>$ $\max_{s_y}$ *s-sim*$^N(s_x, s_y)$ **then**
    tag s_x as literal
else
    tag s_x as nonliteral
end if

precision · recall) / (precision + recall). Nonliteral precision and recall are defined similarly. Average precision is the average of literal and nonliteral precision; similarly for average recall. For overall performance, we take the f-score of average precision and average recall.

We calculated two baselines for each word. The first was a simple majority-rules baseline (assign each word to the sense which is dominant which is always literal in our dataset). Due to the imbalance of literal and nonliteral examples, this baseline ranges from 60.9% to 66.7% for different verbs with an average of 63.6%. Keep in mind though that using this baseline, the f-score for the nonliteral set will always be 0% – which is the problem we are trying to solve in this work. We calculated a second baseline using a simple attraction algorithm. Each sentence in the target set is attracted to the feedback set with which it has the most words in common. For the baseline and for our own model, sentences attracted to neither, or equally to both sets are put in the opposite cluster to which they belong. This second baseline obtains a f-score of 29.36% while the weakly supervised model without active learning obtains an f-score of 53.8%. Results for each verb are shown in Figure 1.

3 Active Learning

The model described thus far is weakly supervised. The main proposal in this paper is to push the results further by adding in an active learning component, which puts the model described in Section 2 in the position of helping a human expert with the literal/nonliteral clustering task. The two main points to consider are: *what* to send to the human annotator, and *when* to send it.

We always send sentences from the undecided cluster – i.e. those sentences where attraction to either feedback set, or the absolute difference of the two attractions, falls below a given threshold. The number of sentences falling under this threshold varies considerably from word to word, so we additionally impose a predetermined cap on the number of sentences that can ultimately be sent to the human. Based on an experiment on a held-out set separate from our target set of sentences, sending a maximum of 30% of the original set was determined to be optimal in terms of eventual accuracy obtained. We impose an order on the candidate sentences using similarity values. This allows the original sentences with the least similarity to either feedback set to be sent to the human first. Further, we alternate positive similarity (or absolute difference) values and values of zero. Note that sending examples that score zero to the human may not help attract new sentences to either of the feedback sets (since scoring zero means that the sentence was not attracted to any of the sentences). However, human help may be the only chance these sentences have to be clustered at all.

After the human provides an identification for a particular example we move the sentence not only into the correct cluster, but also into the corresponding feedback set so that other sentences might be attracted to this certifiably correctly classified sentence.

The second question is when to send the sentences to the human. We can send all the examples after the first iteration, after some intermediate iteration, distributed across iterations, or at the end. Sending everything after the first iteration is best for counteracting false attractions before they become entrenched and for allowing future iterations to learn from the human decisions. Risks include sending sentences to the human before our model has had a chance to make potentially correct decision about them, counteracting any saving of effort. (Karov & Edelman, 1998) state that the results are not likely to change much after the third iteration and we have confirmed this independently: similarity values continue to change until convergence, but cluster allegiance tends not to. Sending everything to the human after the third iteration could therefore entail some of the damage control of sending everything after the first iteration while giving the model

a chance to do its best. Another possibility is to send the sentences in small doses in order to gain some bootstrapping benefit at each iteration i.e. the certainty measures will improve with each bit of human input, so at each iteration more appropriate sentences will be sent to the human. Ideally, this would produce a compounding of benefits. On the other hand, it could produce a compounding of risks. A final possibility is to wait until the last iteration in the hope that our model has correctly clustered everything else and those correctly labeled examples do not need to be examined by the human. This immediately destroys any bootstrapping possibilities for the current run, although it still provides benefits for iterative augmentation runs (see Section 4).

A summary of our results in shown in Figure 1. The last column in the graph shows the average across all the target verbs. We now discuss the various active learning experiments we performed using our model and a human expert annotator.

3.1 Experiment 1

Experiments were performed to determine the best time to send up to 30% of the sentences to the human annotator. Sending everything after the first iteration produced an average accuracy of 66.8%; sending everything after the third iteration, 65.2%; sending a small amount at each iteration, 60.8%; sending everything after the last iteration, 64.9%. Going just by the average accuracy, the first iteration option seems optimal. However, several of the individual word results fell catastrophically below the baseline, mainly due to original sentences having been moved into a feedback set too early, causing false attraction. This risk was compounded in the distributed case, as predicted. The third iteration option gave slightly better results (0.3%) than the last iteration option, but since the difference was minor, we opted for the stability of sending everything after the last iteration. These results show an improvement of 11.1% over the model from Section 2. Individual results for each verb are given in Figure 1.

3.2 Experiment 2

In a second experiment, rather than letting our model select the sentences to send to the human, we selected them randomly. We found no significant difference in the results. For the random model to outperform the non-random one it would have to select only sentences that our model would have clustered incorrectly; to do worse it would have to select only sentences that our model could have handled on its own. The likelihood of the random choices coming exclusively from these two sets is low.

3.3 Experiment 3

Our third experiment considers the effort-savings of using our literal/nonliteral identification model. The main question must be whether the 11.1% accuracy gain of active learning is worth the effort the human must contribute. In our experiments, the human annotator is given at most 30% of the sentences to classify manually. It is expected that the human will classify these correctly and any additional accuracy gain is contributed by the model. Without semi-supervised learning, we might expect that if the human were to manually classify 30% of the sentences chosen at random, he would have 30% of the sentences classified correctly. However, in order to be able to compare the human-only scenario to the active learning scenario, we must find what the average f-score of the manual process is. The f-score depends on the distribution of literal and nonliteral sentences in the original set. For example, in a set of 100 sentences, if there are exactly 50 of each, and of the 30 chosen for manual annotation, half come from the literal set and half come from the nonliteral set, the f-score will be exactly 30%. We could compare our performance to this, but that would be unfair to the manual process since the sets on which we did our evaluation were by no means balanced. We base a hypothetical scenario on the heavy imbalance often seen in our evaluation sets, and suggest a situation where 96 of our 100 sentences are literal and only 4 are nonliteral. If it were to happen that all 4 of the nonliteral sentences were sent to the human, we would get a very high f-score, due to a perfect recall score for the nonliteral cluster and a perfect precision score for the literal cluster. If none of the four nonliteral sentences were sent to the human, the scores for the nonliteral cluster would be disastrous. This situation is purely hypothetical, but should account for the fact that 30 out of 100 sentences annotated by a human will not necessarily result in an average f-score of 30%: in fact, averaging the results of the three sitatuations described above results

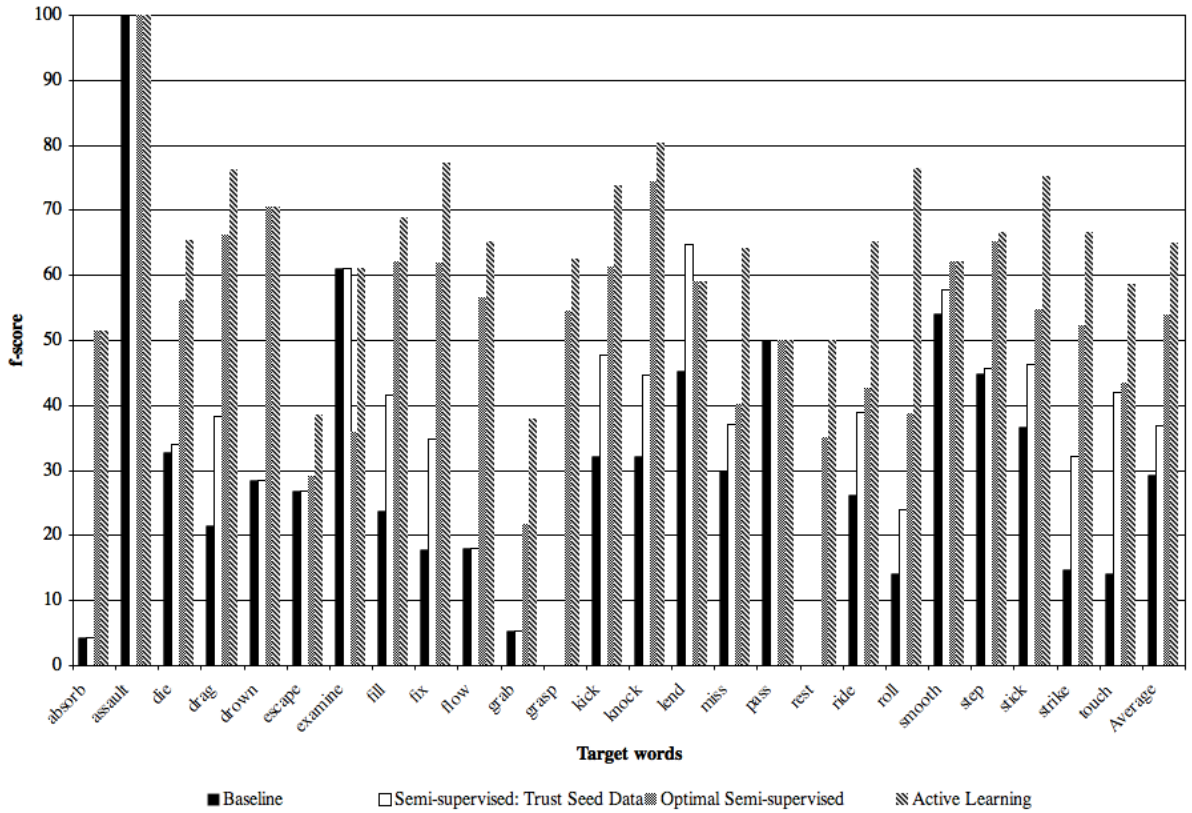

Figure 1: Active Learning evaluation results. *Baseline* refers to the second baseline from Section 2. *Semi-supervised: Trust Seed Data* refers to the standard KE model that trusts the seed data. *Optimal Semi-supervised* refers to the augmented KE model described in (Birke & Sarkar, 2006). *Active Learning* refers to the model proposed in this paper.

in an avarage f-score of nearly 36.9%. This is 23% higher than the 30% of the balanced case, which is 1.23 times higher. For this reason, we give the human scores a boost by assuming that whatever the human annotates in the manual scenario will result in an f-score that is 1.23 times higher. For our experiment, we take the number of sentences that our active learning method sent to the human for each word – note that this is not always 30% of the total number of sentences – and multiply that by 1.23 – to give the human the benefit of the doubt, so to speak. Still we find that using active learning gives us an avarage accuracy across all words of 64.9%, while we get only 21.7% with the manual process. This means that for the same human effort, using the weakly supervised classifier produced a threefold improvement in accuracy. Looking at this conversely, this means that in order to obtain an accuracy of 64.9%, by a purely manual process, the human would have to classify nearly 53.6% of the sentences, as opposed to the 17.7% he needs to do using active learning. This is an effort-savings of about 35%. To conclude, we claim that our model combined with active learning is a helpful tool for a literal/nonliteral clustering project. It can save the human significant effort while still producing reasonable results.

4 Annotated corpus built using active learning

In this section we discuss the development of an annotated corpus of literal/nonliteral usages of verbs in context. First, we examine *iterative augmentation*. Then we discuss the structure and contents of the annotated corpus and the potential for expansion.

After an initial run for a particular target word, we have the cluster results plus a record of the feedback sets augmented with the newly clustered sentences.

*****pour*****
nonliteral cluster
wsj04:7878 N As manufacturers get bigger , they are likely to pour more money into the battle for shelf space , raising the ante for new players ./.
wsj25:3283 N Salsa and rap music pour out of the windows ./.
wsj06:300 U Investors hungering for safety and high yields are pouring record sums into single-premium , interest-earning annuities ./.
literal cluster
wsj59:3286 L Custom demands that cognac be poured from a freshly opened bottle ./.

Figure 2: Excerpt from our annotated corpus of literal/nonliteral usages of verbs in context.

Each feedback set sentence is saved with a *weight*, with newly clustered sentences receiving a weight of 1.0. Subsequent runs may be done to augment the initial clusters. For these runs, we use the the output identification over the examples from our initial run as feedback sets. New sentences for clustering are treated like a regular target set. Running the algorithm in this way produces new clusters and a re-weighted model augmented with newly clustered sentences. There can be as many runs as desired; hence *iterative augmentation*.

We used the iterative augmentation process to build a small annotated corpus consisting of the target words from Table 1, as well as another 25 words drawn from the examples of previously published work (see Section 5). It is important to note that in building the annotated corpus, we used the Active Learning component as described in this paper, which improved our average f-score from 53.8% to 64.9% on the original 25 target words, and we expect also improved performance on the remainder of the words in the annotated corpus.

An excerpt from the annotated corpus is shown in Figure 2. Each entry includes an ID number and a Nonliteral, Literal, or Unannotated tag. Annotations are from testing or from active learning during annotated corpus construction. The corpus is available at *http://www.cs.sfu.ca/∼anoop/students/jbirke/*. Further unsupervised expansion of the existing clusters as well as the production of additional clusters is a possibility.

5 Previous Work

To our knowledge there has not been any previous work done on taking a model for literal/nonliteral language and augmenting it with an active learning approach which allows human expert knowledge to become part of the learning process.

Our approach to active learning is similar to the Uncertainty Sampling approach of (Lewis & Gale, 1994) and (Fujii et. al., 1998) in that we pick those examples that we could not classify due to low confidence in the labeling at a particular point. We employ a resource-limited version in which only a small fixed sample is ever annotated by a human. Some of the criticisms leveled against uncertainty sampling and in favor of Committee-based sampling (Engelson & Dagan, 1996) (and see refs therein) do not apply in our case.

Our similarity measure is based on two views of sentence- and word-level similarity and hence we get an estimate of appropriate identification rather than just correct classification. As a result, by embedding an Uncertainty Sampling active learning model within a two-view clustering algorithm, we gain the same advantages as other uncertainty sampling methods obtain when used in bootstrapping methods (e.g. (Fujii et. al., 1998)). Other machine learning approaches that derive from optimal experiment design are not appropriate in our case because we do not yet have a strong predictive (or generative) model of the literal/nonliteral distinction.

Our machine learning model only does identification of verb usage as literal or nonliteral but it can be seen as a first step towards the use of machine learning for more sophisticated metaphor and metonymy processing tasks on larger text corpora. Rule-based systems – some using a type of interlingua (Russell, 1976); others using complicated networks and hierarchies often referred to as *metaphor maps* (e.g. (Fass, 1997; Martin, 1990; Martin, 1992) – must be largely hand-coded and generally work well on an enumerable set of metaphors or in limited domains. Dictionary-based systems use existing machine-readable dictionaries and path lengths between words as one of their primary sources for metaphor processing information (e.g. (Dolan, 1995)). Corpus-based systems primarily extract or learn the necessary metaphor-processing information from large corpora, thus avoiding the need for manual annotation or metaphor-map construction. Examples of such systems are (Murata et. al., 2000; Nissim & Markert, 2003; Mason, 2004).

Nissim & Markert (2003) approach metonymy resolution with machine learning methods, "which [exploit] the similarity between examples of conventional metonymy" ((Nissim & Markert, 2003), p. 56). They see metonymy resolution as a classification problem between the literal use of a word and a number of pre-defined metonymy types. They use similarities between *possibly metonymic words* (PMWs) and known metonymies as well as context similarities to classify the PMWs.

Mason (2004) presents CorMet, "a corpus-based system for discovering metaphorical mappings between concepts" ((Mason, 2004), p. 23). His system finds the selectional restrictions of given verbs in particular domains by statistical means. It then finds metaphorical mappings between domains based on these selectional preferences. By finding semantic differences between the selectional preferences, it can "articulate the higher-order structure of conceptual metaphors" ((Mason, 2004), p. 24), finding mappings like LIQUID→MONEY.

Metaphor processing has even been approached with connectionist systems storing world-knowledge as probabilistic dependencies (Narayanan, 1999).

6 Conclusion

In this paper we presented a system for separating literal and nonliteral usages of verbs through statistical word-sense disambiguation and clustering techniques. We used active learning to combine the predictions of this system with a human expert annotator in order to boost the overall accuracy of the system by 11.1%. We used the model together with active learning and iterative augmentation, to build an annotated corpus which is publicly available, and is a resource of literal/nonliteral usage clusters that we hope will be useful not only for future research in the field of nonliteral language processing, but also as training data for other statistical NLP tasks.

References

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: an approach to almost parsing. *Comput. Linguist.* 25, 2 (Jun. 1999), 237-265.

Julia Birke and Anoop Sarkar. 2006. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL-2006*. Trento, Italy. April 3-7.

Barbara Di Eugenio and Michael Glass. 2004. The kappa statistic: a second look. *Comput. Linguist.* 30, 1 (Mar. 2004), 95-101.

William B. Dolan. 1995. Metaphor as an emergent property of machine-readable dictionaries. In Proceedings of Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity (March 1995, Stanford University, CA). AAAI 1995 Spring Symposium Series, 27-29.

Sean P. Engelson and Ido Dagan. 1996. In *Proc. of 34th Meeting of the ACL*. 319–326.

Dan Fass. 1997. *Processing metonymy and metaphor*. Greenwich, CT: Ablex Publishing Corporation.

Atsushi Fujii, Takenobu Tokunaga, Kentaro Inui and Hozumi Tanaka. 1998. Selective sampling for example-based word sense disambiguation. *Comput. Linguist.* 24, 4 (Dec. 1998), 573–597.

Yael Karov and Shimon Edelman. 1998. Similarity-based word sense disambiguation. *Comput. Linguist.* 24, 1 (Mar. 1998), 41-59.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proc. of SIGIR-94*.

James H. Martin. 1990. *A computational model of metaphor interpretation*. Toronto, ON: Academic Press, Inc.

James H. Martin. 1992. Computer understanding of conventional metaphoric language. *Cognitive Science* 16, 2 (1992), 233-270.

Zachary J. Mason. 2004. CorMet: a computational, corpus-based conventional metaphor extraction system. *Comput. Linguist.* 30, 1 (Mar. 2004), 23-44.

Masaki Murata, Qing Ma, Atsumu Yamamoto, and Hitoshi Isahara. 2000. Metonymy interpretation using *x no y* examples. In *Proceedings of SNLP2000* (Chiang Mai, Thailand, 10 May 2000).

Srini Narayanan. 1999. Moving right along: a computational model of metaphoric reasoning about events. In *Proceedings of the 16th National Conference on Artificial Intelligence and the 11th IAAI Conference* (Orlando, US, 1999). 121-127.

Malvina Nissim and Katja Markert. 2003. Syntactic features and word similarity for supervised metonymy resolution. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (ACL-03) (Sapporo, Japan, 2003). 56-63.

Adwait Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference* (University of Pennsylvania, May 17-18 1996).

Sylvia W. Russell. 1976. Computer understanding of metaphorically used verbs. *American Journal of Computational Linguistics*, Microfiche 44.