Focusing Annotation for Semantic Role Labeling

Daniel Peterson, Martha Palmer, Shumin Wu
University of Colorado
{daniel.w.peterson, mpalmer, shumin.wu}@colorado.edu

Abstract

Annotation of data is a time-consuming process, but necessary for many state-of-the-art solutions to NLP tasks, including semantic role labeling (SRL). In this paper, we show that language models may be used to select sentences that are more useful to annotate. We simulate a situation where only a portion of the available data can be annotated, and compare language model based selection against a more typical baseline of randomly selected data. The data is ordered using an off-the-shelf language modeling toolkit. We show that the least probable sentences provide dramatic improved system performance over the baseline, especially when only a small portion of the data is annotated. In fact, the lion's share of the performance can be attained by annotating only 10-20% of the data. This result holds for training a model based on new annotation, as well as when adding domain-specific annotation to a general corpus for domain adaptation.

Keywords: Semantics, Language Modeling, Annotation, Semantic Role Labeling, Domain Adaptation

1. Introduction

Annotation of data is a time-consuming process, but necessary for supervised machine learning approaches. Most state-of-the-art solutions to NLP tasks, including semantic role labeling (SRL), are driven by supervised machine learning algorithms. This requires a large amount of annotation to be performed by humans, often by specially-trained linguists. Unfortunately, there are not enough annotation hours available to annotate large amounts of data in every potential domain, and so there is considerable attention paid to increasing the usefulness of annotation efforts. Approaches range from one-shot data ranking approaches (Dligach and Palmer, 2009), where there is no need to iterate between annotation and training, to active learning systems (Lewis and Gale, 1994), where the system is supplied with annotation for the examples it is least confident in, to a combination of these two approaches (Dligach and Palmer, 2011). The foremost approach is taken here.

There are a few advantages to using a one-shot data ranking approach. First, it is simple - find out how much data can be annotated given the resources available, and select that much data to annotate. Second, it makes good use of the annotators' time. Active learning is built on the premise of a back-and-forth iteration between training a model and annotation, and any delays associated with training are realized in lost productivity. Third, as will be shown, it takes only a small portion of the data to get the lion's share of the performance.

2. Contributions

The contributions of this paper are two-fold. First, it demonstrates that language models may be used to select a subset of sentences for annotation, outperforming random selection by a considerable margin on the semantic role labeling task. Second, it shows that this result holds when selecting annotation for adapting a general model to a new domain.

3. Semantic Role Labeling

The semantic role labeling task is recognizing and labeling semantic arguments of a predicate. Typical semantic arguments include *Agent*, *Patient*, *Theme*, etc. and also adjunctive arguments indicating time, location, manner, etc.

Of the many semantic representations (FrameNet, VerbNet, etc), PropBank (Palmer et al., 2005) is the most popular for supervised machine learning approaches because of the wealth of human-annotated corpora. PropBank is layered on top of a constituent-based syntactic parse (Penn Treebank). It annotates verb predicates (and more recently, nominal predicates, adjective predicates, and light-verb constructions) (Bonial et al., 2014) and uses a set of core (numbered) argument and adjunct argument labels on the constituents. `ARG0` typically identifies the *Agent*, while `ARG1` represents *Patient* or *Theme*.

PropBank semantic roles are used in this work. A few example sentences with labeled arguments can be found in Table 1.

(`ARG0` John) **ate** (`ARG1` the fish.)
(`ARG1` The window) **broke**.
(`ARG0` Kate) **threw** (`ARG1` the ball) (`ARG2` over the plate.)

Table 1: Example sentences with semantic role labels. Relations are in bold.

4. Language Modeling for Data Selection

In Dligach and Palmer (2009), it was shown that using only a portion of the training data available was useful for the word sense disambiguation (WSD) task. Because in WSD, most of the training examples are of the most-frequent sense, it is difficult to train accurate models for infrequent senses. The data was ordered using probabilistic language models, from the least probable sentences to the most. The intuition behind this ordering is that low-probability sentences are more likely to contain the low-probability senses of a particular word. This heuristic provided a more balanced training set, with more examples of the infrequent senses relative to the frequent ones.

In Dligach and Palmer (2011), this approach was coupled with active learning. The least probable 10% of the available data was treated as a "seed" set for a standard ac-

tive learning paradigm. A model was trained to perform WSD on the seed set, and then run on the remaining available data. It reported which examples the classifier had the least confidence in. These examples were added, and the model was re-trained, in an iterative fashion. The best performance was achieved using only 15% of the total data for training. This "least probable" seed data significantly outperformed randomly selected seed data, when the same active learning procedure was followed afterward. Using the low-probability sentences ensured that the seed set contained examples of low-probability word senses, that may not be selected by a random seed set.

In this paper, we test whether the same technique may be applicable to the SRL task. Intuitively, the most unusual sentences are more likely to contain the low-probability structures that are important to include in the SRL training data. Uncommon arguments or unusual grammatical structures are likely to appear in low-probability sentences. To organize the sentences, we use the SRI Language Modeling Toolkit (SRILM) (Stolcke, 2002), a free, off-the-shelf toolkit. We trained N-gram language models on our annotated data, and then used those language models to compute the probability of each sentence. This probability score is used to rank sentences from least to most probable. We do not explore active learning in the initial iteration of this system, but this is likely to provide additional benefit.

5. Data Selection for SRL

The experiment was run using ClearSRL (Wu and Palmer, 2011), a state-of-the-art semantic role labeling system, with off-the-shelf settings. This package uses the LIBLINEAR (Fan et al., 2008) classifier for identifying and labeling each argument, relying on constituency-parsed sentences. A corpus of manually-annotated data, roughly 150,000 words, was selected for training. These sentences were ordered using SRILM, off-the-shelf, from least probable to most probable. Testing was carried out on another section of same-domain data, roughly 53,000 words, unseen in training.

The examples in Table 2 show representatives of low-probability and high-probability sentences in the data. In general, the very high-probability sentences are simpler and shorter sentences, with only one or two semantic participants and a single clause. Compound and complex sentences are much harder to label accurately, and in general these sentences have a lower probability.

5.1. Training a model on annotated data

We trained several models, each on only a portion of the available annotated training data. Because the sentences were organized from least to most probable, we could select the "least-probable $n\%$" of the data. For comparison, we also trained models on randomly-selected data, and the "most-probable $n\%$" of the data. On all selection criteria, started with 10% of the data, and added more in 10% increments. The results are summarized in Figure 1, and the data is also in Table 3.

It is clear that the least-probable sentences are more valuable for training. In this simple-to-implement paradigm, the first 10% of the data provides a great deal of the overall performance of the system, beating a system trained on a randomly-selected 30%. Classification accuracy does increase as each successive section of training data is added, but there is only a 3% difference in F1 score between the model trained on the least probable 10% of the sentences, and the model trained on all sentences.

Low probability sentences
"We can force him to produce the poppies illegally and feed into the illegal drug market or we can buy the poppies from him and help provide pain-killing drugs."
"No restaurant I've worked in (and there have been quite a few, ranging from Subway to fine dining) would have found that kind of language acceptable, especially within earshot of customers."
"A consultant in the newspaper article claims that about $12M in private donations is needed annually to support a performance center like Overture, and that a metropolitan area of our size and demographics can be expected to generate about $5M."
High probability sentences
"The law regarding musical copyrights are clear."
"I think they should keep the monarchy."
"I have a question."
"Keep government away from the internet."

Table 2: Example sentences

5.2. Domain adaptation

Because SRL is a well-studied task, there is annotated data available for training. However, if this training data is not similar enough to the target domain, a general model may be unsuitable. In this circumstance, domain adaptation is appropriate. We test a general model trained on OntoNotes (Weischedel et al., 2011), a roughly 1.5 million word corpus of annotated sentences from the Wall Street Journal, broadcast news, newswire text, and several other domains. To this annotated data, we add in annotations from the "least-probable $n\%$" of the data, as above, and retrain the model on all the selected data. The results are summarized in Figure 2. Again we include comparison against randomly-selected data (the same random selection as before), and the "most-probable" data.

The case for low-probability sentences as intelligent training examples is again quite pronounced, but it is worth making a few remarks. First, the baseline model trained on OntoNotes outperforms the best model from the previous step. This is reasonable, given the OntoNotes corpus is an order of magnitude larger. Adding in domain-specific annotation does increase this performance, but the total gain in F1 score from adding the new corpus amounts to only about 1%. Over 40% of the available gain is achieved using only the least-probable 10% of the new data. Exact figures can be found in Table 4.

In these experiments, almost all selection paradigms show a monotonic increase in performance as data is added, so it may seem strange that the trend breaks in domain adaptation, as illustrated clearly in Figure 2. When we add least-probable 70%, 80%, and 100% of the in-domain data, the performance drops.This irregularity may be dependent on the particular corpus, but seems to occur only when most

Training data	Precision	Recall	F1
ALL	85.793391	83.971452	84.872645
10% LP	82.871741	80.724960	81.784265
20% LP	83.866973	81.768667	82.804529
30% LP	84.434066	82.460895	83.435816
40% LP	84.843246	82.957259	83.889654
50% LP	85.173117	83.460331	84.308026
60% LP	85.374313	83.547530	84.451044
70% LP	85.463381	83.752784	84.599436
80% LP	85.645809	83.893644	84.760672
90% LP	85.754990	83.982185	84.859329
10% RND	76.996317	75.723753	76.354733
20% RND	79.625340	78.861581	79.241621
30% RND	81.560082	80.738376	81.147149
40% RND	82.858349	81.634515	82.241879
50% RND	84.118755	82.710418	83.408642
60% RND	84.638847	83.016286	83.819715
70% RND	84.850391	83.236296	84.035594
80% RND	85.085227	83.571678	84.321661
90% RND	85.345075	83.860106	84.596074
10% MP	51.640560	57.450833	54.390967
20% MP	64.582863	65.290709	64.934857
30% MP	72.770033	74.093799	73.425951
40% MP	76.972734	77.901049	77.434109
50% MP	80.431618	80.047490	80.239094
60% MP	81.913629	81.324622	81.618063
70% MP	83.272293	82.356256	82.811741
80% MP	84.240025	83.072630	83.652255
90% MP	85.119592	83.691073	84.399288

Table 3: Scores for SRL models trained on various portions of the training data. LP refers to models where the least probable data was used, RND means random data was selected, MP means the most probable data was used. ALL means all data was used; this result is constant regardless of which data selection paradigm is followed.

Training data	Precision	Recall	F1
ON	87.618718	85.024550	86.302144
ON+ALL	88.512270	85.791903	87.130858
ON+10%LP	88.058401	85.441764	86.730352
ON+20%LP	88.195982	85.510182	86.832318
ON+30%LP	88.305509	85.648359	86.956640
ON+40%LP	88.373992	85.679214	87.005742
ON+50%LP	88.410385	85.789220	87.080082
ON+60%LP	88.415713	85.813367	87.095105
ON+70%LP	88.437124	85.787878	87.092359
ON+80%LP	88.428334	85.744949	87.065971
ON+90%LP	88.525634	85.801293	87.142176
ON+10%RND	87.867723	85.228462	86.527972
ON+20%RND	87.970320	85.408226	86.670342
ON+30%RND	88.050419	85.465912	86.738917
ON+40%RND	88.179731	85.537013	86.838270
ON+50%RND	88.288874	85.561160	86.903618
ON+60%RND	88.377563	85.688605	87.012315
ON+70%RND	88.415284	85.707386	87.040279
ON+80%RND	88.392129	85.748974	87.050492
ON+90%RND	88.451273	85.803976	87.107516
ON+10%MP	87.807812	85.196265	86.482328
ON+20%MP	87.836699	85.262000	86.530201
ON+30%MP	87.955891	85.282123	86.598373
ON+40%MP	88.512270	85.791903	86.699858
ON+50%MP	88.119579	85.453838	86.766238
ON+60%MP	88.267298	85.534330	86.879326
ON+70%MP	88.390198	85.600064	86.972760
ON+80%MP	88.430496	85.691288	87.039346
ON+90%MP	88.443922	85.742266	87.072142

Table 4: Scores for SRL models trained on OntoNotes data, plus various portions of the domain-specific training data. LP refers to models where the least probable data was used, RND means random data was selected, MP means the most probable data was used. ON means OntoNotes was used, and ALL means all domain-specific data was used; these results are constant regardless of which data selection paradigm is followed.

of the data is annotated and added in. The goal of this work is to dramatically reduce the annotation load in a simple one-shot ranking, so even with this oddity the main result remains the same. A more thorough investigation is left for future work.

6. Results

The results in this study are promising. Although for one-shot data ranking, performance generally increases as we add more data, there is a strong desire to reduce the amount of annotation required. We demonstrate that low probability in a language model sense is worthwhile as a proxy for usefulness of data points as training examples. This suggests that the method of Dligach and Palmer (2011), that couples this language model ranking with active learning, can be applied to SRL.

Because compound sentences are likely to be low-probability, it is possible that some of the benefit from selecting low-probability senses is a direct result of the additional number of training clauses. However, if this were the only benefit to selecting the low-probability sentences, it is unlikely that there would be such an improvement over randomly-selected data. We require three times the amount of annotated data to get the same performance from random selection; to get this result only from extra annotated clauses, we would have to expect three times as many clauses per low-probability sentence as there are in average sentences in the corpus. This may account for some of the difference, but there is still a clear advantage to using language model selection.

7. Future Directions

There has been work on active learning in the SRL task (Chen et al., 2011). In (Dligach and Palmer, 2011), active learning showed a considerable benefit from using language modeling to select the seed set of sentences for WSD. Here, the performance increase from language model selection is so drastic, it seems that this result is likely to hold for the SRL task. Along this line, Pradhan et al. (2005) has successfully applied active sampling to the SRL task. While this differs from active learning in that the labels are

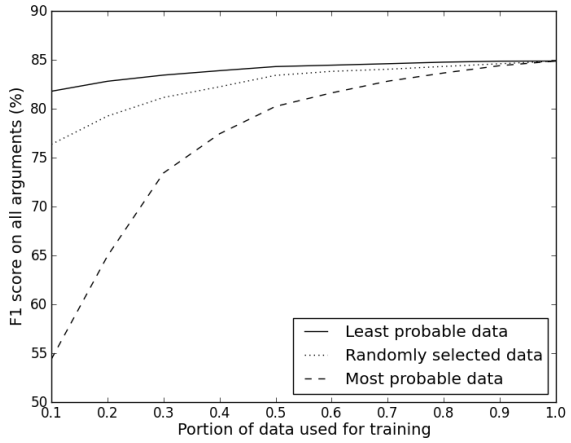

Figure 1: Overall SRL performance as the amount of available training data increases. The data is added from least probable to most probable sentences.

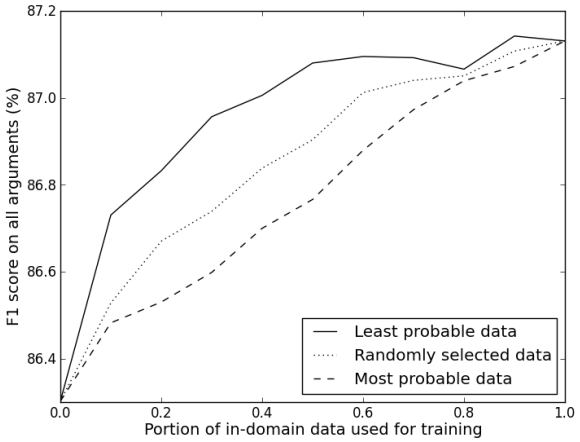

Figure 2: Overall SRL performance on domain adaptation. The base model is always included, and we include portions of the domain-specific training data. The data is added from least probable to most probable sentences.

already known, it demonstrates that, for automatic SRL, a small set of training data can generate a high quality model, especially when combined with the SVM classifier (where only the "support" samples affect the model produced by the learning algorithm).

In addition to providing direction for active learning, the work in (Chen et al., 2011) suggests another compelling experiment, even if an active learning paradigm is not used. The authors use collapsed dependency trees to estimate the "representativeness" of particular training sentences, which balances the tendency of active learning systems to overfit to outlier data points. This same technique could be coupled with language model selection, to hopefully produce an even higher-quality selection of initial data. There does seem to be a real increase in performance when the low-probability sentences are selected, but at least some of these sentences will be low-probability because they are not representative of the data as a whole. This could introduce a bias, that the current experiment does not provide an adequate test for.

Further, it was noted that low-probability sentences are more likely to be long and complex constructions. Compound sentences provide multiple example clauses to train on, per sentence. At least some of the performance increase in this paper is likely to come from these extra examples. Also, these complex sentences take longer to annotate than shorter constructions. However, the low-probability sentences also contain a larger-than-average percentage of rare arguments and modifiers (`ARGM-TMP`, for example, which adds time information to a clause). Although it will require a new annotation effort, it is certainly worth investigating the benefit of language model selection in terms of performance per hour of completed annotation. Based on the results in this paper, it is quite reasonable to expect that language model selection is still a useful ranking scheme to select data for annotation.

8. Conclusions

The experiments in this paper demonstrate that preselecting data using sentence probabilities is promising for the SRL task, in addition to the WSD task. The model performs better on limited training data when this heuristic is employed. Although the results in this paper are only a pilot study, they are quite promising; a larger investigation with more data and more varied domains is justified. Also, further experiments could strengthen the results by demonstrating how much time can be saved with this approach, instead of looking only at how many sentences can be skipped.

Intelligent data selection is not necessarily solved by this approach, and other selection criteria may also be useful to explore. In particular it is worth exploring this technique in conjunction with active learning, like in Dligach and Palmer (2011).

9. Acknowledgements

We gratefully acknowledge the support of DARPA HR0011-11-C-0145 (via LDC) BOLT. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

10. References

Bonial, C., Bonn, J., Conger, K., Hwang, J., and Palmer, M. (2014). Propbank: Semantics of new predicate types. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference*, Reykjavik, Iceland.

Chen, C., Palmer, A., and Sporleder, C. (2011). Enhancing active learning for semantic role labeling via compressed dependency trees. In *Proceedings of International Joint Conference on Natural Language Processing*.

Dligach, D. and Palmer, M. (2009). Using language modeling to select useful annotation data. In *Proceedings of the Student Research Workshop and Doctoral Consortium Held in Conjunction with NAACL-HLT*, Boulder, CO.

Dligach, D. and Palmer, M. (2011). Good seed makes a good crop: Accelerating active learning using language modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, OR.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*.

Lewis, D. and Gale, W. (1994). A sequential algorithm for training text classiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM/Springer.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics Journal*, 31.

Pradhan, S., Hacioglu, K., Ward, W., Martin, J. H., and Jurafsky, D. (2005). Semantic role chunking combining complementary syntactic views. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*.

Stolcke, A. (2002). Srilm – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO.

Weischedel, R., Hovy, E., Marcus, M., Palmer, M., Belvin, R., Pradan, S., Ramshaw, L., and Xue, N., (2011). *OntoNotes: A Large Training Corpus for Enhanced Processing*, pages 54–63. Springer Verlag.

Wu, S. and Palmer, M. (2011). Semantic mappingusing automatic word alignment and semantic role labeling. In *Proceedings of ACL Workshop on Syntax and Structure in Statistical Translation (SSST-5)*, Portland, OR.