LA-UR- 07-0618

Approved for public release;
distribution is unlimited.

Title:	Learning Semantics-Enhanced Language Models Applied to Unsupervised WSD
Author(s):	Karin Verspoor, z#191215, CCS-3 Shou-de Lin, National Taiwan University
Intended for:	Association for Computational Linguistics annual meeting Prague, Czechoslovakia June 25-27, 2007

Form 836 (7/06)

Learning Semantics-Enhanced Language Models Applied to Unsupervised WSD

Abstract

An N-gram language model aims at capturing statistical syntactic word order information from corpora. Although the concept of language models has been applied extensively to handle a variety of NLP problems with reasonable success, the standard model does not incorporate semantic information, and consequently limits its applicability to semantic problems such as word sense disambiguation. We propose a framework that integrates semantic information into the language model schema, allowing a system to exploit both syntactic and semantic information to address NLP problems. Furthermore, acknowledging the limited availability of semantically annotated data, we discuss how the proposed model can be learned without annotated training examples. Finally, we report on a case study showing how the semantics-enhanced language model can be applied to unsupervised word sense disambiguation with promising results.

1 Introduction

Syntax and semantics are two major aspects of language use. Syntax refers to the grammatical structure of a language whereas semantics refers to the meaning of the symbols arranged with that structure. To fully comprehend a language, a human must understand its syntactic structure, the meaning each symbol represents, and the interaction between the two. In most languages, syntactic structure conveys something about the semantics of the symbols, and the semantics of symbols may constrain valid syntactic realizations. As a simple example, when we see a noun following a number in English (e.g. "*one book*"), we can infer that the noun is countable. Conversely, if it is known that a noun is countable, a speaker of English knows that it can plausibly be preceded by a numeral. It is therefore reasonable to assume that for a computer system to successfully process natural language, it has to be equipped with capabilities to represent and utilize both the syntactic and semantic information of the language simultaneously.

The n-gram language model (LM) is a powerful and popular framework for capturing the word order information of language, or fundamentally syntactic information. It has been applied successfully to a variety of NLP problems such as machine translation, speech reorganization, and optical character recognition. As described in equation (1), an n-gram language model utilizes conditional probabilities to capture word order information, and the validity of a sentence can be approximated by the accumulated probability of the successive n-gram probabilities of its constituent words $W_1 \ldots W_k$.

(1) $validity(W_1 W_2 \ldots W_k) = \prod_{i=1}^{k} P(W_i \mid W_{i-n+1} \ldots W_{i-1})$

As powerful as a traditional n-gram LM can be, it does not capture the *semantic* information of a language. Therefore it has seldom been applied to semantic problems such as word sense disambiguation (WSD). To address this limitation, in this paper we propose to expand the formulation of a LM to include not only the words in the sentences but also their semantic labels (e.g. word senses). By incorporating semantic information into a LM, the framework is applicable to problems such as WSD, semantic role labeling, and even more generally machine translation and information extraction – tasks that require both semantic and syntactic information to be solved.

In the next section, we will present our semantics-enhanced language model and discuss an unsupervised method to learn it from untagged text. Section 3 discusses a case study in applying this model for unsupervised WSD. We address the related work in section 4 and conclude section 5.

Deleted: n

Deleted: can therefore be

Formatted: Style ACL Text Indent + (Asian)

Deleted: incorporated

Deleted: the plain

2 Incorporating and Learning Semantics in a Language Models

The first part of this section proposes a semantics-enhanced language model framework while the second part discusses how its parameters can be learned without annotated data.

Formatted: Font: Not Italic

2.1 A semantics-enhanced language model

Figure 1(a) is a general finite state representation of a sentence of four words (W1..W4) connected through a bigram LM. Each word can be regarded as a state node and the transition probabilities between states can be modeled as the n-gram conditional probabilities of the involved states (here we assume the transition probabilities are bigrams). In fact each word in a sentence has a certain lexical meaning (sense or semantic label, S*i*) as represented in Figure 1(b). Conceptually, for each word-based finite state representation there is a dual representation in the semantics (or sense) domain, as shown in 1(c). A *semantics-based LM* (or SLM) like 1(c) records the order relations between meanings. Alternatively, one can combine both representations into a hybrid language model that captures both the word order information and the word meaning, as demonstrated in 1(d). 1(d) represents a *word-sense LM* (or WSLM),

a semantics-enhanced LM incorporating two types of states: word symbols and their semantic labels. The intuition behind WSLM is that when processing a word, people first try to recognize its meaning (i.e. $P(S_n|W_n)$), and based on that predict the next word (i.e. $P(W_{n+1}|S_n)$). Figure 1(e) is the same as 1(d) except that the bigram probabilities are replaced by trigrams. It embodies the concept that the next word to be revealed depends on the previous word together with its semantic label, and the meaning of the current word depends on not only the word itself but the meaning of the previous word.

The major reason for the success of a LM-based approach to NLP problems is its capability of predicting the validity of a sentence. In 1(a), we can say that a sentence "W1W2W3W4" is valid because P(W2|W1)*P(W3|W2)*P(W4|W3) is relatively high. Similarly given that the semantic labels of each word in the sentence are known, the probabilities P(S2|S1)*P(S3|S2)*P(S4|S3) can be applied to assess the semantic validity of this sentence as well. Furthermore, we can say that a word sequence together with its semantic assignment (interpretation) is valid based on a WSLM if the probability of P(S1|W1)*P(W2|S1)*...*P(W4|S3)*P(S4|W4) is high. We can therefore use a semantics-enhanced LM to rank possible interpretations for a word sequence.

2.2 Unsupervised Parameter Learning for a Semantics-enhanced Language Model

The n-gram probabilities of a typical LM such as the transition probabilities in Figure 1(a) can be easily learned through counting term frequencies and co-occurrences from large corpora. If there were some large corpora with semantically annotated words and sentences, we could learn the semantics-enhanced LM such as 1(c) -1(e) directly through frequency counting as well. Unfortunately, there is no corpus containing a significant amount of semantically annotated data available. To address this problem, we discuss below an approach that allows the system to approximate the n-gram probabilities of the semantics-enhanced language models. Without loss of generality, in the following we assume the transition probabilities to be learned are all bigrams.

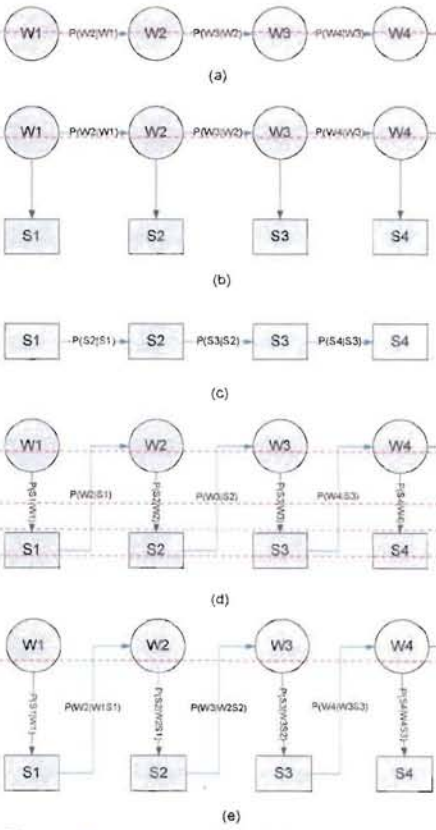

Figure 1: (a) a typical finite-state, bigram LM representation of a sentence. (b) each word in the sentence has a certain meaning (or semantic label). (c) a semantics-enhanced LM, which applies the standard LM directly to semantics. (d) a hybrid LM integrating word and sense information. (e) like (d) except that a trigram model is used.

2.2.1 Learning bigrams for SLM and WSLM

The problem setup is as follows: the system is given a plain-text, unannotated corpus together with a dictionary (assuming WordNet 2.1) that contains a list of plausible semantic labels for each word. Using these resources alone, the system must learn the n-gram dependencies between semantic labels. Note that every word in the WordNet dictionary has at least one sense (or synset label), and each sense has a unique 8-digit id representing its database location. Different words can share synsets, indicating they have senses in common. For example, the word 'result' has four senses in the dictionary and one of these (id=07192761) is shared by the word

Deleted: specific kind of
Deleted: to
Deleted:
Deleted:
Deleted: is
Deleted:
Deleted: .
Deleted:
Formatted: Font: Bold
Deleted: 1

'outcome'. The word 'demonstrate' has four meanings where one of them (id=00656725) also is associated with the words 'prove' and 'show'. To learn a SLM, one has to learn the conditional probabilities of one sense following the other such as $P(S_t=00656725|S_{t-1}=07192761)$.

The first step of learning is to construct a sense-based graph representation for the plain-text corpus by connecting all the plausible senses of each word to the senses of the subsequent word[1]. For example, Figure 2(a) is the sense-graph of the phrase "Existing results demonstrate..." The graph shows there are 3, 4, 4 possible meanings for 'existing', 'result', and 'demonstrate', respectively. The links in the graph, based on the concept of a LM, can be modeled by the n-gram (e.g. bigram) probabilities. If all the bigrams between senses in the graph are known, then for each plausible path of senses (where a path contains one sense per word) we can generate its associated probability, as in equation (2).

(2) ***validity*****(Existing=00965972,results=1124606, demonstrate=021290541)=Pr(00965972|start)* Pr(11246064|00965872)*Pr(02129054|11246064)**

This probability reflects the cumulative validity of each sense assignment for the sequence of words. One can rank all the sense paths based on their probabilities to find the optimal assignment of senses to words. On the other hand, if the associated probability for each path in the graph is given, we can apply a technique called "fractional counting" to determine bigram probabilities. Fractional counting counts the occurrence of each bigram in all possible paths, where the count is weighted by the associated probability of the path.

Unfortunately, without a sense-annotated corpus neither the sense bigrams nor the path probabilities can be known directly. However, since computing the likelihood for each path and generating the bigram probabilities are dual problems (i.e. one can be generated if the other is known), it is possible to apply the expectation-maximization (EM) algorithm (Dempster et al. 1977) to approximate both numbers.

To perform the EM learning, the first step is to initialize the probabilities of the bigrams. As will be shown in our case study, the initialization can be uniformly distributed or use certain preexisting knowledge. In the maximization stage of the EM algorithm, the system uses the initial bigram probabilities to generate the associated probability of each path, such as the one shown in equation (2). In the expectation stage the system applies fractional counting to refine the bigram probabilities. The E-step and M-step continue to iterate until a local optimum (i.e. a path that possesses a locally optimal probability) is reached.

One potential problem for this approach is efficiency. The total number of paths in the graph grows exponentially with the number of words (i.e. $O(b^n)$, where n is the number of words and b is the average branching factor of nodes, i.e. the average number of senses per word). Therefore it is computationally prohibitive for the system to enumerate all paths and produce their associated probabilities one by one to perform fractional counting. Fortunately in this situation one can apply a polynomial forward-backward algorithm (Baum 1972) for fractional counting. Rather than generating all paths with their probabilities in the graph, we need to know only the total probability of all the paths that a link (bigram) occurs in. This can be generated by recording dynamically for each link the accumulated probabilities from the beginning of the graph (the *alpha* value) to the link and the accumulated probabilities from the link to the end (the *beta* value). Since in our case the alpha and beta values are independent, it is possible to generate all n-grams with polynomial time $O(nb^2)$ and space $O(nb)$. A similar approach has been used successfully in other unsupervised NLP problems such as decipherment and machine translation (Cutting et al. 1992; Koehn et al. 2000; Knight et al. 2006; Lin et al. 2006).

The simple example shown in Figure 2 describes the intuition behind the method. Imagine the system encounters the phrases "Existing results demonstrate...", "Existent outcomes show...", "Existing outcomes prove..." in the corpus. According to Figure 2 there is one common sense 00965972 for the words 'existing' and 'existent', a common sense 07192761 for 'results' and 'outcomes'[2], and a single common sense 00656725 for the words demonstrate, show, and prove. Based on the minimum description length principle (or Occam's Razor), a reasonable hypothesis is that these three senses should have higher chance to appear successively compared with the other candidates, since one can then use only three senses to "explain" all the sentences.

[1] Note that the WordNet dictionary contains only nouns, verbs, adjectives and adverbs. Therefore we treat other words such as stop words and proper nouns as having a single sense.

[2] These two words also share sense *11246064*; for the purposes of the example we will ignore this second sense, as given these three phrases alone both senses are equally probable and the choice is arbitrary.

Deleted: , which
Formatted: Font: Bold
Formatted: Font: Bold, Italic
Formatted: Font: Bold
Deleted:
Deleted: ¶
Deleted:
Formatted: Font: Bold
Formatted: Indent: First line: 11.35 pt
Deleted:
Formatted: Font: Italic
Deleted: (the accumulated probabilities from the beginning
Formatted: Indent: First line: 11.35 pt
Deleted: (the accumulated probabilities to the end
Deleted: for each link
Formatted: Font: Italic
Deleted: only
Deleted: ¶
Deleted: S
Deleted: applied
Deleted: to

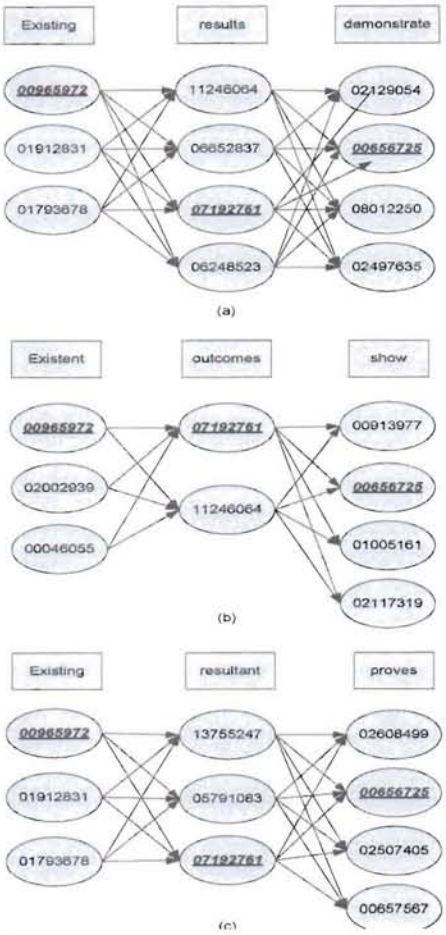

Figure 2: Sense-based graph of word sequences (a) Existing results demonstrate…(b) Existent outcomes show…(c) Existing outcomes prove…

The proposed learning algorithm captures the spirit of this idea. Assuming the initialization stage assigns equal probability to each bigram and assuming all senses listed in Figure 2 do not appear elsewhere in the corpus, then after the 1st iteration of EM, 00656725 will have a higher chance to follow 07192761 compared with others (e.g. equation (3)). This is because the system sees 00656725 following 07192761 more times than others in the fractional counting stage.

(3) Pr(00656725|07192761) > Pr(00913977|07192761)

This approach works because there are situations in which multiple words can be used to express a given meaning, and people tend not to choose the same word repeatedly. The system can take advantage of this to learn information about senses that tend to go together from the shared senses of these varied words, as formalized in the semantics-enhanced LM. The same approach can be applied to learn the parameters in a WSLM as well. The only difference is that the words are included in the graph as single-sense nodes. Figure 3 is the graph presentation of a WSLM.

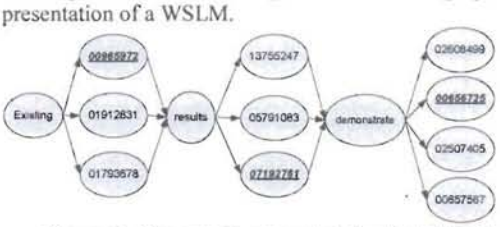

Figure 3: The graph generated for the WSLM. Such a network has the format word1→sense1→ word2→sense2…etc

3 Case Study: Unsupervised Word Sense Disambiguation using SLM and WSLM

In this section, we describe a case study on applying the SLM and WSLM to perform an all-words word sense disambiguation task. Since both language models are trained without sense-annotated data, this task is an unsupervised WSD task.

3.1 Background

Unsupervised WSD aims at determining the senses of words in a text without using a sense-annotated corpus for training. The methods employed generally fall into two categories, one for all-words, token-based WSD (i.e. assign each token a sense in its individual sentential context) and the other to find most frequent sense of each unique token in the text as a whole (following a "one sense per discourse" assumption). The motivation to focus on the second type of task is that assigning the most frequent sense to every word turns out to be a simple heuristic that outperforms most approaches (Hoste et al. 2002). The following paragraphs describe the existing unsupervised WSD methods.

Banerjee and Pedersen proposed a method that exploits the concept of gloss overlap for WSD (Banerjee et al. 2003). It assumes the sense whose gloss definition looks most similar (i.e. overlap strongly) with the glosses of surrounding content words is the correct one. Mihalcea's graph-based algorithm (Mihalcea 2005) first constructs a weighted

Formatted: Font: Bold

Deleted: can be described as

Deleted: the

Formatted: Caption, Justified

Deleted: ¶

Deleted:

Deleted: see

Formatted: Font: 11 pt, Bold

Formatted: Font: Bold

sense-based graph[3], where weights are the similarity between senses (e.g. gloss overlap). Then it applies PageRank to identify prestigious senses as the correct interpretation. Galley and McKeown also propose a graph-based approach called lexical chains that regards a sense to be dominant if it has more strong connections with its context words (Galley et al. 2003). The strength of connection is determined by the type of relation as well as the distance between the words in the text. Navigli and Velardi propose a conceptually similar but more knowledge-intensive approach called structural semantic interconnections (SSI) (Navigli et al. 2005). For each sense, the method first constructs the semantic graphs consisting of collocation information (extracted from annotated corpora), WordNet relation information, and domain labels. Using these graphs, the algorithm iteratively chooses senses with strong connectivity to the relevant senses in the semantic graph as the correct ones. McCarthy et al. propose a method to determine the most frequent senses for words (McCarthy et al. 2004). In their framework, the distributionally similar neighbors of each word are determined, and a sense of a word is regarded as dominant if it is the most similar to the senses of its distributionally similar neighbors.

Although the above methods try to tackle the unsupervised WSD problem from different angles, they do share a common theme of identifying the sense that is *semantically* the most "similar" or "related" to the context or neighbor words as the correct one. The WordNet or other dictionary relations as well as their similarity measure play important roles in the disambiguation. While we are not arguing the legitimacy of this strategy, we believe there is another type of information that a system can benefit from to determine the sense of words, namely the word and sense order information encoded in a LM. Based on this alternative strategy even the non-content words such as stop words (ignored in existing approaches) can be helpful. Considering the sentence *"He went into the bank beside the river"*, most of the above approaches will likely choose the *"river bank" (bank#2)* sense for *bank* instead of the correct *"financial institute" (bank#1)* sense, because the former sense is semantically closer to the only other function word "river". However, even without other context information, it is not hard for an English speaker to realize the financial bank is more likely to be the correct one, since people do not usually go *into* a river bank. A somewhat accurate SLM can guide the system to make this decision since it shows

$$P(bank\#1|into\#1\,the\#1) >> P(bank\#2|into\#1\,the\#1).$$

Such information can be learned in an unsupervised manner if the system gets a chance to see other similar sentences such as *"he went into a banking-company"* (where *banking-company* has *bank#1* sense in WordNet 2.1). Also consider the sentence "The tank has an empty tank". Again it is not trivial for the previously described algorithms to realize these two *tanks* have different meanings since their frameworks (explicitly or implicitly) imply or result in one sense per sentence. However, an accurate semantics-enhanced language model can tell us that the *tank as container* sense has higher chance to follow the word *empty* while the *tank as the army tank* sense has higher chance to be followed by *has*.

3.2 System Design and Experiment setup

We applied both bigram SLM and WSLM to perform unsupervised WSD. Our WSD system can be divided into three stages. The first stage is the initialization stage. In SLM, we need to initialize $P(S_{k+1}|S_k)$ and in WSLM there are two types of probabilities to be initialized: $P(S_k|W_k)$ and $P(W_{k+1}|S_k)$. We designed four different ways to initialize the LMs with or without the preliminary knowledge. The second stage is the learning stage, using the EM algorithm together with forward-backward training to learn the bigrams. The final stage is the decoding stage, in which the learned bigrams are utilized to identify the senses of words in their sentential context that optimize the total probability. Using the dynamic programming method, the overall time complexity for the system is only linear to the number of words and quadratic to the average number of senses per word.

We tested our system on SemCor (SC) data, which is a sense-annotated corpus that contains a total of 778K words (where 234K have sense annotations). We use SemCor and British National Corpus (BNC) sampler data (1.1 million words) for training[4]. The experimental setup is as follows: we first determine the baseline performance on the WSD task using only the initial knowledge (i.e. without applying language models). Then we train a semantics-enhanced LM based on the initialization and use it to perform

[3] The graph is similar to Figure 1(c), except that it ignores the words with single sense such as the stop words, which we believe to be useful in disambiguation.

[4] The annotated senses in SemCor were not used for training. Log probabilities are used throughout the training and decoding stage to prevent overflow.

Deleted: the
Deleted: find the
Deleted: one
Deleted: the
Deleted: of them
Deleted: -

decoding. Our model is evaluated by checking how much the learned LM can *improve* the accuracy.

3.3 Initializing without knowledge

In the first two types of initialization no external knowledge other than the unannotated corpus and the sense dictionary is exploited. The baseline for this case is a random sense assignment for all-words WSD (i.e. disambiguation of all word tokens) in SemCor, resulting in 17% accuracy on the test set.

The first initialization simply assigns equal probability to all bigrams. As shown in Table 1 (Uniform initialization), the results go up to 32.3% for SLM and 28.8% for WSLM after training on a corpus consisting of the SemCor texts plus texts from the BNC Sampler.

Initialization	Corpus	Baseline (%) without LM	SLM (%)	WSLM (%)
Uniform	SC	17.1	**31.8**	**27.7**
Uniform	SC+BNC	17.1	**32.3**	**28.8**
Graph freq	SC	17.1	**35.1**	**34.0**
Graph freq	SC+BNC	17.1	**36.0**	**34.6**

Table 1: The results for all-words unsupervised WSD on SemCor using SLM and WSLM based on uniform and node-frequency initialization.

The second initialization is based on the node occurrence frequency in the sense graph. That is, $Pr(S1|S2) \propto gf(S1)$ for SLM and $Pr(S1|W1) \propto gf(S1)$ for WSLM[5], where gf(S1) represents the frequency of a node S1 in the sense graph, or its *graph frequency* (for example, in Figure 2 '*00965972*' appears three times). The intuition behind this initialization is that a sense should have a higher chance to appear if it occurs in multiple words that frequently occur in the text. Again, to count the node frequency we do not need any extra knowledge since the graph itself can be generated based on only the corpus and the dictionary. This initialization improves the accuracy to 36.0% for SLM and 34.6% for WSLM. These initializations tell us that the learned syntactic order structure can tell us much about the senses of the words in context, in the absence of additional external knowledge.

3.4 Initializing based on Distributional and WordNet Similarity Score

The EM algorithm is known for its sensitivity to the initialization for unsupervised NLP tasks (see examples in (Knight et al. 2006)). Therefore how our LM is initialized can affect the final results significantly. As we have described it, the LM for WSD does not take advantage of lexical semantic information, which has been shown to be useful by other systems such as SSI and McCarthy's method. Fortunately, an important advantage of our framework is that it is possible to incorporate existing knowledge by using it to initialize the model. We therefore performed an experiment exploiting McCarthy's sense scores (here called M-score) to initialize the bigram probabilities, and then applying our EM learning engine to refine the model with the hope of improving the final results. For example, for WSLM, we assign initial $P(Sk|Wk) \propto$ M-score(Sk,Wk), and for SML, we initialize $P(Sk+1|Sk) \propto$ M-score(Sk,Wk)+M-score(Sk+1, Wk+1). We use Lin's distributional similarity score (Lin 1998) together with Jiang and Cornath's WordNet similarity measure (Jiang et al. 1998) to generate an M-score for the senses of nouns and verbs[6]. The baseline in this case is established by choosing the senses with the highest M-score, which reaches 45.1%[7] for all words WSD in SemCor. As shown in Table 2, our system can improve the WSD results by 3-4%. This experiment demonstrates how our framework allows one to take advantage of both the syntactic (LM) and semantic (distributional WordNet similarity) information to get improved results for an unsupervised WSD task.

3.5 Initializing based on Sense Order

The final initialization assigns initial bigram probability based on the frequency order of senses provided in WordNet. Specifically, we initialize the bigram Pr(W1#x|W2#y) frequency as $1/(x+y+1)$, assuming x and y represents the rank of the senses based on their frequency as defined in WordNet. For example, in SLM $Pr(\text{bank\#2}|\text{into\#1}) = \frac{1}{2+1+1} = 1/4$, given bank#2 represents the 2nd most frequent sense of *bank* and into#1 represents the most frequent sense of *into*. For WSLM, $Pr(\text{bank\#2}|\text{bank}) = \frac{1}{2+1} = 1/3$ (the word does not contribute to the initialization). The baseline uses WordNet sense frequency order to assign the most frequent sense to each token, which reaches 69% accuracy on all words WSD for SemCor. This accuracy is regarded as the upper

[5] Pr(W2|S1) is uniformly distributed for WSLM

[6] Only scores for verbs and nouns are produced here since Jiang&Conrath similarity does not generate similarity scores for other parts of speech.

[7] The senses for the words that do not have M-score are chosen randomly.

Deleted: using

Formatted: Font: 6 pt

Deleted: ¶

Formatted: Font: 2 pt

bound for predominant sense assignment. After using the refined LM model learned by EM to perform WSD, the WSLM result goes up to 71% (See table 2). This result shows that by adding word and sense order information to WordNet sense frequency order, we are able to perform slightly better than the upper bound of the predominant sense assignment. This result is encouraging in that as we know there is no other existing unsupervised WSD system that can surpass this upper bound.

Initialization	Corpus	Baseline (%) without LM	SLM (%)	WSLM (%)
M-Score	SC	45.1	**48.2**	**48.3**
M-Score	SC+BNC	45.1	**48.4**	**48.6**
WNOrder	SC	69.2	**69.7**	**71.2**
WNOrder	SC+BNC	69.2	**69.3**	**71.1**

Table 2: The results for all-words unsupervised WSD on SemCor using SLM and WSLM based on M-score and WordNet synset order initialization.

3.6 Discussion

The case study on applying semantics-enhanced LM to WSD reveals two important facts. The first is that syntactic order information for words and senses can benefit WSD. This conclusion to some extent echoes the concept of syntactic semantics (Rapaport 2002), which claims that semantics are embedded inside syntax. The second conclusion is that the unsupervised learning method proposed in this paper does learn a sufficient amount of meaningful semantic order information to allow the system to improve disambiguation quality, and it is flexible enough to incorporate existing knowledge through different initializations.

Table 3 shows how different types of knowledge perform in WSD. We compare our system with two existing WSD systems on the all-nouns WSD task (that is, evaluating disambiguation performance only on nouns in the corpus): Banerjee and Pedersen's gross overlap system and the SSI system (results as reported in (Brody et al. 2006)). The LM-based approach without preliminary knowledge performs right in between gross overlap and SSI approaches in predicting the nouns in SemCor. This is interesting and informative since the results demonstrate that by using only word order information and no lexical semantic information (e.g. sense similarity), we still generate competitive WSD results. Moreover, as the bottom two lines of table 3 show, our system is capable of boosting performance when the lexical semantic information is added during initialization. Comparing table 3 with previous tables, one can also infer that WSD on nouns is an easier task than on other parts of speech.

Initialization	Corpus	SLM (%)	WSLM (%)
Uniform	SC+BNC	35.6	32.3
gross overlap		36.5	
Graph freq	SC+BNC	39.6	38
SSI		42.7	
M-Score	SC+BNC	65.0	66.5
WN_order	SC+BNC	75.3	77.4

Table 3: Comparison between LM-based approaches, semantic approaches and semantics-enhanced LM approaches for all-nouns Unsupervised WSD

One advantage of our model is that it can incorporate supervised information as well. A small amount of annotated data can be used to generate the initial n-grams and to be refined through EM. Finally, the framework is flexible enough to be trained on a domain-specific corpus to obtain a SLM or WSLM specifically for that domain.

4 Related Work

There have been previous efforts in incorporating semantics into a language model. Brown et al. proposed a class-based language model that includes semantic classes in a LM (Brown et al. 1992). Bellegarda proposes to exploit latent semantic analysis to map words and their relationships with documents into a vector space (Bellegarda 2000). Chueh et al. propose to combine semantic topic information with n-gram LM using the maximum entropy principle (Chueh et al. 2006). Griffiths et al. also proposes to integrate topic semantic information (Griffiths et al. 2004) into syntax based on a Markov chain Monte Carlo method.

The major difference between our model and these is that we propose to learn semantics at the word level rather than at the document or topic level. Consequently the models are different in the parameters to be learned (in the other models, the topic usually determines words to be used while in our model the words can determine senses), preliminary knowledge incorporation (e.g. Brown et al. used a fully connected word-class mapping during initialization) and most importantly, the applications. Other systems were evaluated on word clustering or document classification while we have made the first attempt to apply a semantics-enhanced LM to a fine-grained semantic analysis task, namely WSD.

Deleted: -

Formatted: Font: Bold

Formatted: Font: 6 pt, Bold

Deleted: the

Deleted: trying to

Deleted: integrat

Deleted: e

Deleted: into an

Deleted: a

Deleted:

Deleted: the

Deleted: s

Deleted: the

Deleted: based on

Deleted: theirs

Deleted: in

Deleted: while their definition of semantics is on

Deleted: e.g.

Deleted: -

Deleted: in

Deleted: evaluation and

Deleted: Most of the

Deleted: o

Deleted: cast

Deleted: on

Deleted:

5 Conclusion and Future Work

There are three major contributions in this paper. First we propose a framework that enables us to incorporate semantics into a language model. Second we show how such model can be learned efficiently ($O(nb^2)$ in time) in an unsupervised manner. Third we demonstrate how this model can be used to perform the WSD task and how additional knowledge can be exploited to improve the performance of the model on the task. Our experiments also suggest that WSD can be a suitable platform to evaluate the semantic language models.

The future directions are two-fold. In terms of the model itself, we would like to investigate how much the results can be improved based on higher n-gram models (e.g. trigram) and investigate how other semantic information (e.g. WordNet hierarchy) can be incorporated into the model. In terms of applications we would like to investigate whether the model can be applied to other NL tasks that generally require both syntactic and semantic information such as information extraction, summarization, and machine translation.

6 References

S. Banerjee and T. Pedersen (2003). "Extended gloss overlaps as a measure of semantic relatedness." In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, p.805-810.

L. E. Baum (1972). "An Inequality and Associated Maximization in Statistical Estimation for Probabilistic Functions of Markov Processes." Inequalities 627(3): p.1-8.

J. Bellegarda (2000). "Exploiting latent semantic information in statistical language modeling." Proceedings of IEEE 88(8): p.1279-1296.

S. Brody, R. Navigli and M. Lapata (2006). "Ensemble Methods for Unsupervised WSD." In Proceedings of the ACL/COLING, p.97-104.

P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai and R. L.Mercer (1992). "Class-based n-gram models of natural language." Computational Linguistics, vol.18, no.4 18(4): p.467 - 479.

C.-H. Chueh, H.-M. Wang and J.-T. Chien (2006). "A Maximum Entropy Approach for Semantic Language Modeling." Computational Linguistics and Chinese Language Processing 11(1): p.37-56.

D. Cutting, J. Kupiec, J. Pedersen and P. Sibun (1992). "A practical part-of-speech tagger." In Proceedings of the In Proceedings of ANLP-92, Trento, Italy, p.133--140.

A. D. Dempster, N. M. Laird and D. B. Rubin (1977). "Maximum likelihood for incomplete data via the EM algorithm." Journal of Royal Statistical Society Series B 39: p.1-38.

M. Galley and K. McKeown (2003). "Improving Word Sense Disambiguation in Lexical Chaining." In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, p.1486-1488.

T. Griffiths, M. Steyvers, D. Blei and J. Tenenbaum (2004). "Integrating Topics and Syntax." In Proceedings of the Advances in Neural Information Processing Systems.

V. Hoste, I. Hendrickx, W. Daelemans and A. Bosch (2002). "Parameter optimization for machine-learning of word sense disambiguation." Language Engineering 8(4): p.311-325.

J. J. Jiang and D. W. Conrath (1998). "Semantic similarity based on corpus statistics and lexical taxonomy." In Proceedings of the International Conference on Research in Computational Linguistics (ROCLING), Taiwan.

K. Knight, A. Nair, N. Rathod and K. Yamada (2006). "Unsupervised Analysis for Decipherment Problems." In Proceedings of the ACL-COLING.

P. Koehn and K. Knight (2000). "Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm." In Proceedings of the AAAI, p.711--715.

D. Lin (1998). "An information-theoretic definition of similarity." In Proceedings of the Proceedings of the 15th ICML, Madison, WI, p.296 - 304.

S.-d. Lin and K. Knight (2006). "Discovering the linear writing order of a two-dimensional ancient hieroglyphic script." Artificial Intelligence 170(4-5).

D. McCarthy, R. Koeling, J. Weeds and J. Carroll (2004). "Finding predominant word senses in untagged text." In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain.

R. Mihalcea (2005). "Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling." In Proceedings of the Joint Conference on Human Language Technology / Empirical Methods in Natural Language Processing (HLT/EMNLP).

R. Navigli and P. Velardi (2005). "Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation." IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27(7)(7): p.1063-1074.

W. J. Rapaport (2002). "Holism, Conceptual-Role Semantics, and Syntactic Semantics." Minds and Machines 12(1): p.3-59.

Deleted: