Language Models Based on Semantic Composition

Jeff Mitchell and **Mirella Lapata**
School of Informatics, University of Edinburgh
Edinburgh EH8 9LW, UK
jeff.mitchell@ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

In this paper we propose a novel statistical language model to capture long-range semantic dependencies. Specifically, we apply the concept of semantic composition to the problem of constructing predictive history representations for upcoming words. We also examine the influence of the underlying semantic space on the composition task by comparing spatial semantic representations against topic-based ones. The composition models yield reductions in perplexity when combined with a standard n-gram language model over the n-gram model alone. We also obtain perplexity reductions when integrating our models with a structured language model.

1 Introduction

Statistical language modeling plays an important role in many areas of natural language processing including speech recognition, machine translation, and information retrieval. The prototypical use of language models is to assign probabilities to sequences of words. By invoking the chain rule, these probabilities are generally estimated as the product of conditional probabilities $P(w_i|h_i)$ of a word w_i given the history of preceding words $h_i \equiv w_1^{i-1}$. In theory, the history could span any number of words up to w_i such as sentences or even a paragraphs. In practice, however, it has proven challenging to deal with the combinatorial growth in the number of possible histories which in turn impacts reliable parameter estimation. A simple and effective strategy is to truncate the chain rule to include only the n-1 preceding words (n is often set within the range of 3–5). The simplification reduces the number of free parameters. However, low values of n impose an artificially local horizon to the language model, and compromise its ability to capture long-range dependencies, such as syntactic relationships, semantic or thematic constraints.

The literature offers many examples of how to overcome this limitation, essentially by allowing the modulation of probabilities by dependencies which extend to words beyond the n-gram horizon. Cache language models (Kuhn and de Mori, 1992) increase the probability of words observed in the history, e.g., by some factor which decays exponentially with distance. Trigger models (Rosenfeld, 1996) go a step further by allowing arbitrary word pairs to be incorporated into the cache. Structured language models (e.g., Roark (2001)) go beyond the representation of history as a linear sequence of words to capture the syntactic constructions in which these words are embedded.

It is also possible to build representations of history which are semantic rather than syntactic (Bellegarda (2000; Coccaro and Jurafsky (1998; Gildea and Hofmann (1999)). In this approach, estimates for the probabilities of upcoming words are derived from a comparison of their semantic content with the content of the history so far. The semantic representations, in this case, are vectors derived from the distributional properties of words in a corpus, based on the insight that words which are semantically similar will be found in similar contexts (Harris, 1968; Firth, 1957). Although the the construction of a semantic representation for the history is crucial to this approach, the underlying vector-based models are primarily designed to represent isolated words rather than word sequences. Ideally, we would like to *compose* the meaning of the history out of its constituent parts. This is by no means a new idea. Much work in linguistic theory (Partee, 1995; Montague, 1974) has been devoted to compositionality, the process of determining the meaning of complex expressions from simpler ones. Previous work either ignores this issue (e.g., Bellegarda (2000)) or simply com-

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 430–439,
Singapore, 6-7 August 2009.

putes the centroid of the vectors representing the history (e.g., Coccaro and Jurafsky (1998)). This is motivated primarily by mathematical convenience rather than by empirical evidence.

In our earlier work (Mitchell and Lapata, 2008) we formulated composition as a function of two vectors and introduced a variety of models based on addition and multiplication. In this paper we apply vector composition to the problem of constructing predictive history representations for language modeling. Besides integrating composition with language modeling, a task which is novel to our knowledge, our approach also serves as a valuable testbed of our earlier framework which we originally evaluated on a small scale verb-subject similarity task. We also investigate how the choice of the underlying semantic representation interacts with the choice of composition function by comparing a spatial model that represents words as vectors in a high-dimensional space against a probabilistic model that represents words as topic distributions.

Our results show that the proposed composition models yield reductions in perplexity when combined with a standard n-gram model over the n-gram model alone. We also show that with an appropriate composition function spatial models outperform the more sophisticated topic models. Finally, we obtain further perplexity reductions when our models are integrated with a structured language model, indicating that the two approaches to language modeling are complementary.

2 Background

2.1 Distributional Models of Semantics

The insight that words with similar meanings will tend to be distributed in similar contexts has given rise to a number of approaches that construct semantic representations from corpora. Broadly speaking, these models come in two flavors. *Semantic space* models represent the meaning of words in terms of vectors, with the vector components being derived from the distributional statistics of those words. Essentially, these models provide a simple procedure for constructing spatial representations of word meaning. *Topic models*, in contrast, impose a probabilistic model onto those distributional statistics, under the assumption that hidden topic variables drive the process that generates words. Both approaches represent the meanings of words in terms of an n-dimensional series of values, but whereas the semantic space model treats those values as defining a vector with spatial properties, the topic model treats them as a probability distribution.

A simple and popular (McDonald, 2000; Bullinaria and Levy, 2007; Lowe, 2000) way to construct a semantic space model is to associate each vector component with a particular context word, and assign it a value based on the strength of its co-occurrence with the target (i.e., the word for which a semantic representation is being constructed). For example, in Mitchell and Lapata (2008) we used the 2,000 most frequent content words in a corpus as their contexts, and defined co-occurrence in terms of the context word being present in a five word window on either side of the target word. We calculated the ratio of the probability of the context word given the target word to the overall probability of the context word and use these values as their vector components. This procedure has the benefits of simplicity and also of being largely free of any additional theoretical assumptions over and above the distributional approach to semantics. This is not to say that more sophisticated approaches have not been developed or that they are not useful. Much work has been devoted to enriching semantic space models with syntactic information (e.g., Grefenstette (1994; Padó and Lapata (2007)), selectional preferences (Erk and Padó, 2008) or with identifying optimal ways of defining the vector components (e.g., Bullinaria and Levy (2007)).

The semantic space discussed thus far is based on word co-occurrence statistics. However, the statistics of how words are distributed across the documents also carry useful semantic information. Latent Semantic Analysis (LSA, Landauer and Dumais (1997) utilizes precisely this distributional information to uncover hidden semantic factors by means of dimensionality reduction. Singular value decomposition (SVD, Berry et al. (1994)) is applied to a word-document co-occurrence matrix which is factored into a product of a number of other matrices; one of them represents words in terms of the semantic factors and another represents documents in terms of the same factors. The algebraic relation between these matrices can be used to show that any document vector is a linear combination of the vectors representing the words it contains. Thus, within this paradigm it is nat-

ural to treat multi-word structures as a "pseudo-document" and represent them via linear combinations of word vectors.

Due to its generality, LSA has proven a valuable analysis tool with a wide range of applications. However, the SVD procedure is somewhat ad-hoc lacking a sound statistical foundation. Probabilistic Latent Semantic Analysis (pLSA, Hofmann (2001)) casts the relationship between documents and words in terms of a generative model based on a set of hidden topics. Documents are represented by distributions over topics and topics are distributions over words. Thus the mixture of topics in any document determines its vocabulary. Maximum likelihood estimation of these distributions over a word-document matrix has a comparable effect to SVD in LSA: a set of hidden semantic factors, in this case topics, are extracted and documents and words are represented by these topics.

Latent Dirichlet Allocation (Griffiths et al., 2007; Blei et al., 2003) enhances further the mathematical foundation of this approach. Whereas pLSA treats each document as a separate, independent mixture of topics, LDA assumes that the topic distributions of documents are generated by a Dirichlet distribution. Thus, LDA is a probabilistic model of the whole document collection. In this model the process of generating a document can be described as follows:

1. draw a multinomial distribution θ from a Dirichlet distribution parametrized by α
2. for each word in a document:
   (a) draw a topic z_k from the multinomial distribution characterized by θ
   (b) draw a word from a multinomial distribution conditioned on the topic z_k and word probabilities β

Under this model, constructing a representation for a multi-word sequence amounts to estimating the topic proportions for that sequence.[1] Structure here arises from the mathematical form of the model, as opposed to any linguistic assumptions.

Without anticipating our results too much, we should point out that several features of the LDA model are likely to affect the representation of multi-word sequences. Firstly, it is a top-down generative model (the topic proportions for a document are first selected and then this drives the generation of words) as opposed to a bottom-up constructive process (words modulate each other to produce a complex representation of their combination). Secondly, the top level Dirichlet distribution is likely to lead to documents being dominated by a small number of topics, producing sparse vectors. And lastly, the assumption that words are generated independently means the interaction between them is not modeled.

[1] Estimating the posterior distribution $P(\theta, z|\mathbf{w}, \alpha, \beta)$ of the hidden variables given an observed collection of documents $\mathbf{w}$ is intractable in general; however, a variety of approximate inference algorithms have been proposed in the literature (e.g., Blei et al. (2003; Griffiths et al. (2007)).

2.2 Language Modeling using Semantic Representations

A common approach to embedding semantic representations within language modeling is to measure the semantic similarity between an upcoming word and its history and use it to modify the probabilities from an n-gram model. In this way, the n-gram's sensitivity to short-range dependencies is enriched with information about longer-range semantic coherence. Much of previous work has taken this approach (Bellegarda, 2000; Coccaro and Jurafsky, 1998; Wandmacher and Antoine, 2007), whilst relying on LSA to provide semantic representations for individual words. Some authors (Coccaro and Jurafsky, 1998; Wandmacher and Antoine, 2007) use the geometric notion of a vector centroid to construct representations of history, whereas others (Bellegarda, 2000; Deng and Khundanpur, 2003) use the idea of a "pseudo-document", which is derived from the algebraic relation between documents and words assumed within LSA. They all derive $P(w_i|h_i)$, the probability of an upcoming word given its history, from the cosine similarity measure which must be somehow normalized in order to yield well-formed probability estimates.

The approach of Gildea and Hofmann (1999) overcomes this difficulty by using representations constructed with pLSA, which have a direct probabilistic interpretation. As a result, the probability of an upcoming word given the history can be derived naturally and directly, avoiding the need for ad-hoc transformations. In constructing their representation of history, Gildea and Hofmann (1999) use an online Expectation Maximization process, which derives from the probabilistic basis of pLSA, to update the history with new words.

Extensions on the basic semantic language

models sketched above involve representing the history by multiple LSA models of varying granularity in an attempt to capture topic, subtopic, and local information (Zhang and Rudnicky, 2002); incorporating syntactic information by building the semantic space over words and their syntactic annotations (Kanejiya et al., 2004); and treating the LSA similarity as a feature in a maximum entropy language model (Deng and Khundanpur, 2003).

3 Composition Models

The problem of vector composition has received relatively little attention within natural language processing. Attempts to use tensor products (Smolensky, 1990; Clark et al., 2008; Widdows, 2008) as a means of binding one vector to another face major computational difficulties as their dimensionality grows exponentially with the number of constituents being composed. To overcome this problem, other techniques (Plate, 1995) have been proposed in which the binding of two vectors results in a vector which has the same dimensionality as its components. Crucially, the success of these methods depends on the assumption that the vector components are randomly distributed. This is problematic for modeling language which has regular structure.

Given the above considerations, in Mitchell and Lapata (2008) we introduce a general framework for studying vector composition, which we formulate as a function f of two vectors $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{h} = f(\mathbf{u}, \mathbf{v}) \tag{1}$$

where $\mathbf{h}$ denotes the composition of $\mathbf{u}$ and $\mathbf{v}$. Different composition models arise, depending on how f is chosen. Our earlier work (Mitchell and Lapata, 2008) explored two broad classes of models based on additive and multiplicative functions.

Additive models are the most common method of vector combination in the literature. They have been applied to a wide variety of tasks including document coherence (Foltz et al., 1998), essay grading (Landauer and Dumais, 1997), modeling selectional restrictions (Kintsch, 2001), and notably language modeling (Coccaro and Jurafsky, 1998; Wandmacher and Antoine, 2007):

$$h_i = u_i + v_i \tag{2}$$

Vector addition (or averaging, which is equivalent under the cosine similarity measure) is a computationally efficient composition model as it does not increase the dimensionality of the resulting vector. However, the idea of averaging is somewhat counterintuitive from a linguistic perspective. Composition of simple elements onto more complex ones must allow the construction of novel meanings which go beyond those of the individual elements (Pinker, 1994).

In Mitchell and Lapata (2008) we argue that composition models based on multiplication address this problem:

$$h_i = u_i \cdot v_i \tag{3}$$

Whereas the addition of vectors 'lumps their content together', multiplication picks out the content relevant to their combination by scaling each component of one with the strength of the corresponding component of the other. This argument is appealing, especially if one is interested in explaining how the meaning of a verb is modulated by its subject. Here, we also develop a complementary, probabilistic argument for the validity of this model.

Let us assume that semantic vectors are based on components defined as the ratio of the conditional probability of a context word given the target word to the overall probability of the context word.

$$v_i = \frac{p(context_i|target)}{p(context_i)} \tag{4}$$

These vectors represent the distributional properties of a given target word in terms of the strength of its co-occurrence with a set of context words. Dividing through by the overall probability of each context word prevents the vectors being dominated by the most frequent context words, which will often also have the highest conditional probabilities.

Let us assume vectors $\mathbf{u}$ and $\mathbf{v}$ represent target words w_1 and w_2. Now, when we compose these vectors using the multiplicative model and the components definition in (4), we obtain:

$$h_i = v_i \cdot u_i = \frac{p(c_i|w_1)}{p(c_i)} \frac{p(c_i|w_2)}{p(c_i)} \tag{5}$$

And by Bayes' theorem:

$$h_i = \frac{p(w_1|c_i)p(w_2|c_i)}{p(w_1)p(w_2)} \tag{6}$$

Assuming w_1 and w_2 are independent and applying Bayes' theorem again, h_i becomes:

$$h_i \approx \frac{p(w_1 w_2|c_i)}{p(w_1 w_2)} = \frac{p(c_i|w_1 w_2)}{p(c_i)} \tag{7}$$

By comparing to (4), we can see that the expression on the right hand side gives us something akin to the vector components we would expect when our target is the co-occurrence of w_1 and w_2. Thus, for the multiplicative model, the combined vector h_i can be thought of as an approximation to a vector representing the distributional properties of the phrase $w_1 w_2$.

If multiplication results in a vector which is something like the representation of w_1 *and* w_2, then addition produces a vector which is more like the representation of w_1 *or* w_2. Suppose we were unsure whether a word token x was an instance of w_1 or of w_2. It would be reasonable to express the probabilities of context words around this token in terms of the probabilities for w_1 and w_2, assuming complete uncertainty between them:

$$p(c_i|x) = \frac{1}{2}p(c_i|w_1) + \frac{1}{2}p(c_i|w_2) \tag{8}$$

Therefore, we could represent x with a vector, based on these probabilities, having the components:

$$x_i = \frac{1}{2}\frac{p(c_i|w_1)}{p(c_i)} + \frac{1}{2}\frac{p(c_i|w_2)}{p(c_i)} \tag{9}$$

Which is exactly the vector averaging approach to semantic composition. As more vectors are combined, vector addition will lead to greater generality rather than greater specificity. The multiplicative approach, on the other hand, picks out the components of the constituents that are relevant to the combination, and represents more faithfully the properties of their conjunction.

As an aside, we should point out that our earlier work (Mitchell and Lapata, 2008) introduced several other models, additive and multiplicative, besides the ones discussed here. We selected the additive model as a baseline and also due to its overwhelming popularity in the language modeling literature. The multiplicative model presented above performed best in our evaluation study (i.e., predicting *verb-subject* similarity).

4 Language Modeling

Estimating Probabilities In language modeling our aim is to derive probabilities, $p(w|h)$, given the semantic representations of word, w, and its history, h, based on the assumption that probable words should be semantically coherent with the history. Semantic coherence is commonly measured via the cosine of the angle between two vectors:

$$sim(\mathbf{w},\mathbf{h}) = \frac{\mathbf{w}\cdot\mathbf{h}}{|\mathbf{w}||\mathbf{h}|} \tag{10}$$

$$\mathbf{w}\cdot\mathbf{h} = \sum_i w_i h_i \tag{11}$$

where $\mathbf{w}\cdot\mathbf{h}$ is the dot product of $\mathbf{w}$ and $\mathbf{h}$. Coccaro and Jurafsky (1998) utilize this measure in their approach to language modeling. Unfortunately, they find it necessary to resort to a number of ad-hoc mechanisms to turn the cosine similarities into useful probabilities. The primary problem with the cosine measure is that, although its values lie between 0 and 1, they do not sum to 1, as probabilities must. Thus, some form of normalization is required. A further problem concerns the fact that such a measure takes no account of the underlying frequency of w, which is crucial for a probabilistic model. For example, *encephalon* and *brain* are roughly synonymous, and may be equally similar to some context, but *brain* may nonetheless be much more likely, as it is generally more common.

An ideal measure would take account of the underlying probabilities of the elements involved and produce values that sum to 1. Our approach is to modify the dot product (equation (11)) on which the cosine measure is based. Assuming that our vector components are given by equation (4), the dot product becomes:

$$\mathbf{w}\cdot\mathbf{h} = \sum_i \frac{p(c_i|w)}{p(c_i)}\frac{p(c_i|h)}{p(c_i)} \tag{12}$$

which we modify to derive probabilities as follows:

$$p(w|h) = p(w)\sum_i \frac{p(c_i|w)}{p(c_i)}\frac{p(c_i|h)}{p(c_i)}p(c_i) \tag{13}$$

This expression now weights the sum with the independent probabilities of the context words and the word to be predicted. That this is indeed a valid probability can be seen by the fact it is equivalent to $\sum_i p(w|c_i)p(c_i|h)$. However, in constructing a representation of the history h, it is more convenient to work with equation (13) as it is based on vector components and can be readily used with the composition models presented in Mitchell and Lapata (2008).

Equation (13) allows us to derive probabilities from vectors representing a word and its prior history. We must also construct a representation of

the history up to the nth word of a sentence. To do this, we combine, via some (additive or multiplicative) function f, the vector representing that word with the vector representing the history up to $n-1$ words:

$$\mathbf{h}_n = f(\mathbf{w}_n, \mathbf{h}_{n-1}) \tag{14}$$

$$\mathbf{h}_1 = \mathbf{w}_1 \tag{15}$$

One issue that must be resolved in implementing equation (14) is that the history vector should remain correctly normalized. In other words, the products $h_i \cdot p(c_i)$ must themselves be a valid distribution over context words. So, after each vector composition the history vector is normalized as follows:

$$h_i = \frac{\hat{h}_i}{\sum_j \hat{h}_j \cdot p(c_i)} \tag{16}$$

Equations (13)–(16) define a language model that incorporates vector composition. To generate probability estimates, it requires a set of word vectors whose components are based on the ratio of probabilities described by equation (4).

Our discussion thus far has assumed a spatial semantic space model similar to that employed in Mitchell and Lapata (2008). However, there is no reason why the vectors should not be constructed by some other means. As mentioned earlier, in the LDA topic model, words are represented as distributions over topics. These distributions are essentially components of a vector $\mathbf{v}$ corresponding to the target word for which we wish to construct a semantic representation. Analogously to equation (4), we convert these probabilities to ratios of probabilities:

$$v_i = \frac{p(topic_i|target)}{p(topic_i)} \tag{17}$$

Integrating with Other Language Models The models defined above are based on little more than semantic coherence. As such they will be only weakly predictive, since they largely ignore word order, which n-gram models primarily exploit. The simplest means to integrate semantic information with a standard language model involves combining two probability estimates as a weighted sum:

$$p(w|h) = \lambda_1 p_1(w|h) + (1-\lambda) p_2(w|h) \tag{18}$$

Linear interpolation is guaranteed to produce valid probabilities, and has been used, for example, to integrate structured language models with n-gram models (Roark, 2001). However, it will work best when the models being combined are roughly equally predictive and have complementary strengths and weaknesses. If one model is much weaker than the other, linear interpolation will typically produce a model of intermediate strength (i.e., worse than the better model), with the weaker model contributing a form of smoothing at best.

Therefore, based on equation (13), we express our semantic probabilities as the product of the unigram probability, $p(w)$, and a semantic component, Δ, which determines the factor by which this probability should be scaled up or down given the context in which it occurs.

$$p(w|h) = p(w) \cdot \Delta(w,h) \tag{19}$$

$$\Delta(w,h) = \sum_i \frac{p(c_i|w)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i) \tag{20}$$

Thus, it seems reasonable to integrate the n-gram model by replacing the unigram probabilities with the n-gram versions.[2]

$$\hat{p}(w_n) = p(w_n|w_{n-2}^{n-1}) \cdot \Delta(w_n, h) \tag{21}$$

To obtain a true probability estimate we normalize $\hat{p}(w_n)$ by dividing through the sum of all word probabilities:

$$p(w_n|w_{n-2}^{n-1}, h) = \frac{\hat{p}(w_n)}{\sum_w \hat{p}(w)} \tag{22}$$

In integrating our semantic model with an n-gram model, we allow the latter to handle short range dependencies and have the former handle the longer dependencies outside the n-gram window. For this reason, the history h used by the semantic model in the prediction of w_n only includes words up to w_{n-3} (i.e., only words outside the n-gram).

We also integrate our models with a structured language model (Roark, 2001). However, in this case we use linear interpolation (equation (18)) because the models are roughly equally predictive and also because linear interpolation is widely used when structured language models are combined with n-grams and other information sources. This approach also has the benefit of allowing the

[2] Equation (21) can also be expressed as $p(w_n|w_{n-2}^{n-1}, h) \approx \frac{p(w_n|w_{n-2}^{n-1}) p(w_n|h)}{p(w_n)}$, Which is equivalent to assuming that h is conditionally independent of w_{n-2}^{n-1} (Gildea and Hofmann, 1999).

models to be combined without out the need to renormalize the probabilities. In the case of the structured language model, normalizing across the whole vocabulary would be prohibitive.

5 Experimental Setup

In this section we discuss our experimental design for assessing the performance of the models presented above. We give details on our training procedure and parameter estimation, and present the methods used for comparison with our approach.

Method Following previous work (e.g., Bellegarda (2000)) we integrated our compositional language models with a standard n-gram model (see equation (21)). We experimented with additive and multiplicative composition functions, and two semantic representations (LDA and the simpler semantic space model), resulting in four compositional models. In addition, we compared our models against a state of the art structured language model in order to assess the extent to which the information provided by the semantic representation is complementary to syntactic structure. Our experiments used Roark's (2001) grammar-based language model. Similarly to standard language models, it computes the probability of the next word based upon the previous words of the sentence. This is done by computing a subset of all possible grammatical relations for the prior words and then estimating the probability of the next grammatical structure and the probability of seeing the next word given each of the prior grammatical relations. When estimating the probability of the next word, the model conditions on the two prior heads of constituents, thereby using information about word triples (like a trigram model).

All our models were evaluated by computing perplexity on the test set. Roughly, this quantifies the degree of unpredictability in a probability distribution, such that a fair k-sided dice would have a perplexity of k. More precisely, perplexity is the reciprocal of the geometric average of the word probabilities and a lower score indicates better predictions.

Parameter Estimation The compositional language models were trained on the BLLIP corpus, a collection of texts from the Wall Street Journal (years 1987–89). The training corpus consisted of 38,521,346 words. We used a development corpus of 50,006 words and a test corpus of similar size. All words were converted to lowercase and numbers were replaced with the symbol $\langle$num$\rangle$. A vocabulary of 20,000 words was chosen and the remaining tokens were replaced with $\langle$unk$\rangle$.

Following Mitchell and Lapata (2008), we constructed a simple semantic space based on co-occurrence statistics from the BLLIP training set. We used the 2,000 most frequent word types as contexts and a symmetric five word window. Vector components were defined as in equation (4). Contrary to our earlier work, we did not lemmatize the corpus before constructing the vectors as in the context of language modeling this was not appropriate. We also trained the LDA model on BLLIP, using Blei et al.'s (2003) implementation.[3] We experimented with different numbers of topics on the development set (from 10 to 200) and report results on the test set with 100 topics. In our experiments, the hyperparameter α was initialized to 0.5, and the β word probabilities were initialized randomly.

We integrated our compositional models with a trigram model which we also trained on BLLIP. The model was built using the SRILM toolkit (Stolcke, 2002) with backoff and Good-Turing smoothing. Ideally, we would have liked to train Roark's (2001) parser on the same data as that used for the semantic models. However, this would require a gold standard treebank several times larger than those currently available. Following previous work on structured language modeling (Roark, 2001; Charniak, 2001; Chelba and Jelinek, 1998), we therefore trained the parser on sections 2–21 of the Penn Treebank containing 936,017 words. Note that Roark's (2001) parser produces prefix probabilities for each word of a sentence which we converted to conditional probabilities by dividing each current probability by the previous one.

6 Results

Table 1 shows perplexity results when the compositional models are combined with an n-gram model. With regard to the simple semantic space model (SSM) we observe that both additive and multiplicative approaches to constructing history are successful in reducing perplexity over the n-gram baseline, with the multiplicative model outperforming the additive one. This confirms the

[3] Available from `http://www.cs.princeton.edu/~blei/lda-c/index.html`.

Model	Perplexity
n-gram	78.72
n-gram+Add$_{\text{SSM}}$	76.65
n-gram + Multiply$_{\text{SSM}}$	75.01
n-gram+Add$_{\text{LDA}}$	76.60
n-gram+Multiply$_{\text{LDA}}$	123.93
parser	173.35
n-gram + parser	75.22
n-gram + parser + Add$_{\text{SSM}}$	73.45
n-gram + parser + Multiply$_{\text{SSM}}$	71.32
n-gram + parser + Add$_{\text{LDA}}$	71.58
n-gram + parser + Multiply$_{\text{LDA}}$	87.93

Table 1: Perplexities for n-gram, composition and structured language models, and their combinations; subscripts $_{\text{SSM}}$ and $_{\text{LSA}}$ refer to the semantic space and LDA models, respectively.

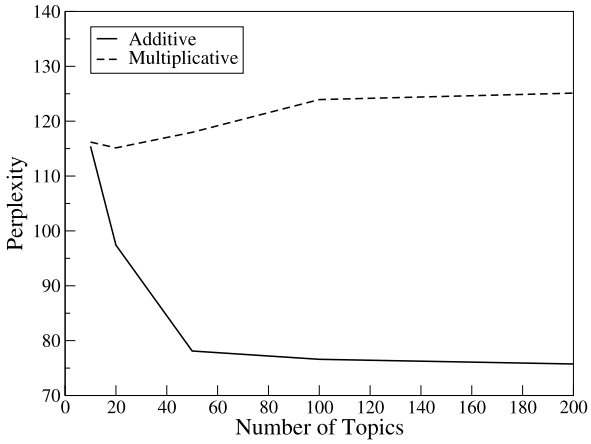

Figure 1: Perplexity versus Number of Topics for the LDA models using additive and multiplicative composition functions.

hypothesis that for this type of semantic space the multiplicative vector combination function produces representations which have a sounder probabilistic basis.

The results for the LDA model are also reported in the table. This model reduces perplexity with an additive composition function, but performs worse than the n-gram with a multiplicative function. For comparison, Figure 1 plots the perplexity of the combined LDA and n-gram models against the number of topics. Increasing the number of topics produces higher dimensional representations which ought to be richer, more detailed and therefore more predictive. While this is true for the additive model, a greater number of topics actually increases the perplexity of the multiplicative model, indicating it has become less predictive.

We compared these perplexity reductions against those obtained with a structured language model. Following Roark (2001), we combined the structured language model with a trigram model using linear interpolation (the weights were optimized on the development set). This model (n-gram + parser) performs comparably to our best compositional model (n-gram + Multiply$_{\text{SSM}}$). While both models incorporate long range dependencies, the parser is trained on a hand annotated treebank, whereas the compositional model uses raw text, albeit from a larger corpus. Interestingly, when interpolating the trigram with the parser *and* the compositional models, we obtain additional perplexity reductions. This suggests that the semantic models are encoding useful predictive information about long range dependencies, which is distinct from and potentially complementary to the parser's syntactic information about such dependencies. Note that the semantic space multiplicative model yields the highest perplexity reduction in this suite of experiments followed by the LDA additive model.

7 Conclusions

In this paper we advocated the use of vector composition models for language modeling. Using semantic representations of words outside the n-gram window, we enhanced a trigram model with longer range dependencies. We compared composition models based on addition and multiplication and examined the influence of the underlying semantic space on the composition task. Our results indicate that the multiplicative composition function produced the most predictive representations with a simple semantic space. Interestingly, its effect in the LDA setting was detrimental. Increasing the representational power of the LDA model, by using a greater number of topics, rendered the multiplicative model less predictive.

These results, together with the basic mathematical structure of the LDA model, suggest that it may not be well suited to forming representations for word sequences. In particular, the assumption that words are generated independently within documents prevents the interactions between words being modeled. This assumption, along with the Dirichlet prior on document distributions tends to lead to highly sparse word vec-

tors, with a typical word being strongly associated with only one or two topics. Multiplication of a number of these vectors generally produces a vector in which most of these associations have been obliterated by the sparse components, resulting in a representation with little predictive power.

These shortcomings arise from the mathematical formulation of LDA, which is not directed at modeling the semantic interaction between words. An interesting future direction would be to optimize the vector components of the probabilistic model over a suitable training corpus, in order to derive a vector model of semantics adapted specifically to the task of composition. We also plan to investigate more sophisticated composition models that take syntactic structure into account. Our results on interpolating the compositional models with a parser indicate that there is substantial mileage to be gained by combining syntactic and semantic dependencies.

Acknowledgements We are grateful to Brian Roark for making his parser available to us. Thanks to Frank Keller and Victor Lavrenko for insightful comments and suggestions. This work was supported by the Economic and Social Research Council [grant number PTA-030-2006-00341] and the Engineering and Physical Sciences Research Council [grant number GR/T04540/01].

References

Jerome R. Bellegarda. 2000. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296.

Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. 1994. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

J.A. Bullinaria and J.P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–123, Toulouse, France.

Ciprian Chelba and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, pages 225–231, Montréal, Canada.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the 2nd Symposium on Quantum Interaction*, pages 133–140, Oxford, UK. College Publications.

Noah Coccaro and Daniel Jurafsky. 1998. Towards better integration of semantic predictors in satistical language modeling. In *Proceedings of the 5th International Conference on Spoken Language Processing*, pages 2403–2406, Sydney, Australia.

Yonggang Deng and Sanjeev Khundanpur. 2003. Latent semantic information in maximum entropy language models for conversational speech recognition. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–63, Edmonton, AL.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Honolulu, Hawaii.

J. R. Firth. 1957. A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*, pages 1–32. Philological Society, Oxford.

Peter Foltz, Walter Kintsch, and Thomas Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Process*, 15:285–307.

Daniel Gildea and Thomas Hofmann. 1999. Topic-based language models using EM. In *Proceedings of the 6th European Conference on Speech Communiation and Technology*, pages 2167–2170, Budapest, Hungary.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA.

Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*, 114(2):211–244.

Zellig Harris. 1968. *Mathematical Structures of Language*. Wiley, New York.

Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 41(2):177–196.

Dharmendra Kanejiya, Arun Kumar, and Surendra Prasad. 2004. Statistical language modeling with performance benchmarks using various levels of

syntactic-semantic information. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1161–1167, Geneva, Switzerland.

Walter Kintsch. 2001. Predication. *Cognitive Science*, 25(2):173–202.

Roland Kuhn and Renato de Mori. 1992. A cache based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (14):570–583.

T. K. Landauer and S. T. Dumais. 1997. A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

Will Lowe. 2000. *Topographic Maps of Semantic Space*. Ph.D. thesis, University of Edinburgh.

Scott McDonald. 2000. *Environmental Determinants of Lexical Processing Effort*. Ph.D. thesis, University of Edinburgh.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, OH.

R. Montague. 1974. English as a formal language. In R. Montague, editor, *Formal Philosophy*. Yale University Press, New Haven, CT.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

B. Partee. 1995. Lexical semantics and compositionality. In Lila Gleitman and Mark Liberman, editors, *Invitation to Cognitive Science Part I: Language*, pages 311–360. MIT Press, Cambridge, MA.

S. Pinker. 1994. *The Language Instinct: How the Mind Creates Language*. HarperCollins, New York.

Tony A. Plate. 1995. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

Roni Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language*, 10:187–228.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, CO.

Tonio Wandmacher and Jean-Yves Antoine. 2007. Methods to integrate a language model with semantic information for a word prediction component. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 506–513, Prague, Czech Republic.

Dominic Widdows. 2008. Semantic vector products: Some initial investigations. In *Proceedings of the 2nd Symposium on Quantum Interaction*, Oxford, UK. College Publications.

Rong Zhang and Alexander I. Rudnicky. 2002. Improve latent semantic analysis based language model by integrating multiple level knowldege. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 893–897, Denver, CO.