A Language Model Sensitive to Discourse Context*

Tae-Gil Noh
Institute für Computerlinguistik
Heidelberg University
69120 Heidelberg, Germany
noh@cl.uni-heidelberg.de

Sebastian Padó
Institute für Maschinelle Sprachverarbeitung
University of Stuttgart
70569 Stuttgart, Germany
pado@ims.uni-stuttgart.de

Abstract

The paper proposes a meta language model that can dynamically incorporate the influence of wider discourse context. The model provides a conditional probability in forms of P(text|context), where the context can be arbitrary length of text, and is used to influence the probability distribution over documents. A preliminary evaluation using a 3-gram model as the base language model shows significant reductions in perplexity by incorporating discourse context.

1 Introduction

Language models (LMs) are designed to distinguish likely from unlikely texts, for example judging that P("I broke my leg") is more likely than P("I ate my leg"). This type of prediction helps various tasks like Speech Recognition and Machine Translation (Pieraccini, 2012; Koehn, 2010).

The most prominent family of LMs in widespread use today is the family of *n-gram models* (Manning and Schütze, 1999; Zweig and Burges, 2012) which model the probability of a word as its conditional probability given the n-1 preceding words, $P(x_n|x_1, \ldots, x_{n-1})$. This assumption makes estimation of the model parameters easy, but the resulting models cannot take into account broader discourse context. For example, consider the two sentences "I broke my hand." and "I broke my promise." According to a standard LM (Brants and Franz, 2006), both are about equally likely to appear in written text. However, if the previous sentence was "I fell from a ladder." a human reader can easily predict that "I broke my hand" is much more likely to follow than "I broke my promise". This cannnot be accounted for straightforwardly within n-gram language models since it would involve raising n to high values.

The method in this paper dynamically incorporates the influence of wider discourse context into a LM which we call the *Conditioned Language Model (CLM)*. It models the influence of context by defining a conditional probability distribution in the form of $P(text|context)$, where both texts and context can be word sequences of arbitrary length. The model builds on the observation that not all documents in a large corpus are equally relevant for a given $text$. Inspired by the use of LMs in Information Retrieval (Manning et al., 2008), we assign weights to corpus documents based on the $context$, in effect giving documents which make the $context$ more likely a higher weight in the scoring of the $text$. For example, using the $context$ "fell from a ladder" would assign higher weight to documents about household accidents and lead to higher probabilities for $text$s like "broke my hand".

The CLM is not a standalone language model, but a meta-model similar to smoothing or domain adaptation methods. It can be applied to any base language model appropriate for LM-based IR. We present an efficient implementation of the CLM and a pilot evaluation on a news corpus with an underlying trigram LM. We find that the CLM can use discourse context to improve predictions for sentences in unseen documents, significantly reducing per-word perplexity compared to the base LM, with the highest reductions for small (i.e., specific) contexts.

2 The Model

2.1 The Query Likelihood Model

Our Conditioned Language Model builds on *document-based language models* as commonly used in Information Retrieval, such as the *query likelihood model* (Ponte and Croft, 1998; Miller et al., 1999; Manning et al., 2008). The query likelihood model constructs a LM M_{d_i} from each document d_i in a corpus. The model scores each document d_i relative to a query q formulated as a set of terms $\{t_1 t_2 \dots t_k\}$ by the conditional probability $P(d_i|q)$ which can be written as

$$P(d_i|q) = P(q|d_i)P(d_i)/P(q) \quad (1)$$

Since $P(q)$ is fixed for a given query and $P(d)$ is often set to the uniform distribution, it is sufficient to optimize $P(q|d_i)$, the probability that a query q would be drawn by random sampling from the document d_i. It is generally computed by assuming that the query decomposes into a sequence of smaller units (terms or n-grams u_k) which can be assumed to be conditionally independent of one another given the document:

$$P(q|d_i) = P(u_1 \dots u_k|d_i) = \prod_k P(u_k|d_i) \quad (2)$$

Finally, the probability of each unit given a document is generally defined as an interpolation of the collection LM and the document LM:

$$P(u_k|d_i) = \lambda P_{M_{d_i}}(u_k) + (1-\lambda)P_{M_C}(u_k) \quad (3)$$

where M_C is a LM trained on the whole collection, while M_{d_i} is a LM just for d_i. This interpolation counteracts sparsity, ensuring that all $P(u_k|d_i)$ are defined over the same events and assign some probability to units even if they do not appear in d_i.

2.2 The Conditioned Language Model

Our Conditioned Language Model (CLM) extends the Query Likelihood Model in a manner that is fairly straightforward when the models are visualized as generative processes, as shown in Figure 1. In the query likelihood model (shown on the left), the document generates the query; in the conditioned language model (on the right-hand side), the document generates both the text and its context.

We assume that context and text are generated using the same process, defined in Eq. (2). The important extension of the CLM is that it allows us to define a conditional probability for a text t given some context c, $P(t|c)$. By marginalizing over documents, it can be defined as:

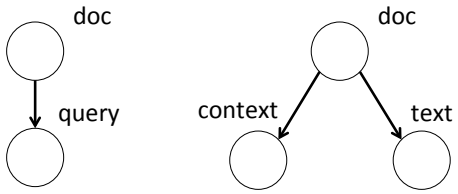

Figure 1: The query likelihood model (left) and the conditioned language model (right)

$$P(t|c) = \frac{P(c,t)}{P(c)} = \frac{\sum_i P(c,t,d_i)}{\sum_i P(d_i,c)} \quad (4)$$

$$= \frac{\sum_i P(d_i)P(c|d_i)P(t|d_i)}{\sum_i P(d_i)P(c|d_i)} \quad (5)$$

Assuming a uniform prior over documents yields:

$$P(t|c) = \sum_i \left(\frac{P(c|d_i)}{\sum_j P(c|d_j)} P(t|d_i) \right) \quad (6)$$

$$= \sum_i P(d_i|c)P(t|d_i) \quad (7)$$

The step from Eq. (6) to Eq. (7) involves an application of Bayes' rule as well as the assumption of a uniform prior over the documents.

Eq. (7) can be used to illustrate the relationship between the traditional LM and the CLM. In a traditional n-gram based LM, the only straightforward ways to incorporate information akin to our context would be to concatenate text and context into a combined query $c + t$. Due to the independence assumptions of the LM , $P(c+t) = P(c)P(t)$.[1] Thus, text and context are independent of each other.

This is fundamentally different in the CLM where text and context are generally *not* independent. If occurrences of t and c are associated (i.e. there are many documents that can generate both c and t), then $P(c,t) > P(c)P(t)$. Conversely, if they are unlikely to be observed together, $P(c,t) < P(c)P(t)$. The reason for this behavior is that even though the model assumes that context

[1] This is true provided that there are no units of the type $\dots c_m$ </s> <s> $t_1 \dots$ "cutting across" the boundary between c and t. We believe that this is a reasonable assumption.

and text are generated independently given the document (as shown in Figure 1), knowing the context can be understood to update the document distribution so that it is non-uniform and conditioned on the context ($P(d_i|c)$). In this way, the CLM assigns to every text t (given c) the probability that the text is generated from a document, where the contribution of the document is weighted by its probability given the context c.

This results in a dynamic LM which can be interesting for a range of applications, by encoding previous knowledge into the context variable. Examples include lexical substitution tasks (McCarthy and Navigli, 2009) or sentence completion tasks that have been specifically articulated as challenges for LM (Zweig and Burges, 2012). The example that we used in the introduction can also be phrased as such a problem: *I fell from a ladder and broke my* ***hand / promise / heart***.

3 Efficiency Considerations

Querying the CLM for a text t involves calling every document LM to compute $P(t|d_i)$, which is potentially expensive. To improve efficiency, we can take advantage of the fact that $P(t|d_i)$ is typically very non-uniformly distributed: only a very small number of documents are highly relevant for a given text. To assess this effect, we have experimented with retrieving just the top N documents. We index all documents with the Apache Solr[2] search engine and retrieve the first N documents returned by a Boolean search for the query t.[3] We set the document-based probability term $P_{M_{d_i}}$ from $P(t|d_i)$ from Eq. (3) to 0 for all documents that are not returned. This cuts off the "long tail" of the document-based distribution part of $P(t|d_i)$. We find that setting N to 10,000 typically captures 99% of the total probability mass of $P(t|d_i)$ and yields quasi-optimal performance.

Calculating $P(t|d_i)$ for large Ns of documents (e.g., 10,000) seems like a serious time complexity overhead. However, it is not necessary to actually call the document LMs N times. In fact, the Query Likelihood model is generally used to produce a document ranking, for which task it also needs to compute $P(query|d_i)$ for all d_i. Implementations solve this task efficiently by keeping inverted indices that not only record document IDs, but also the probability of n-grams for each document model. Such index structures can be very large, but provide near real-time calculations of $P(query|d_i)$ on large document sets. The same strategies can be used to compute the two terms comprising the CLM (cf. Eq. (7)), with just a constant overhead (computing two terms instead of one for each document plus a weighted sum).

[2] `http://lucene.apache.org/solr/`

[3] The documents are ranked by the number of matching terms. Ties are broken randomly.

4 Experimental Setup

We presents a pilot evaluation using per-word perplexity, a standard task-independent proxy for improvements in language modeling. Perplexity be understood as the amount of information necessary to encode the text, with lower numbers indicating better models.

Language Model. We construct our base LM from the 1.6M AFP news articles (700M words) from English Gigaword corpus (Parker et al., 2011) using SRILM (Stolcke, 2002). The collection model M_C is trained one the complete corpus. Document n-gram LMs M_{d_i} are generated from each document. All models are trained using a standard setup: trigrams with Katz back-off (Katz, 1987) and Good-Turing smoothing (Gale and Sampson, 1995). The CLM is implemented as described in Eq. (6) and Section 3.

Baselines. We consider two baselines. The first one is the collection model P_{M_C} which does not use any document models. $P_{\text{CLM}}(s_n|\emptyset)$ is the CLM without context. This model corresponds to the Query Likelihood Model (Eq. (3)), assuming a uniform distribution over documents.

Test and validation data. We use a set of 50 news articles from APW February 2010 for the optimization of the interpolation parameter λ. The final test evaluation takes place on the unseen first 500 news articles of Gigaword APW January 2010 subcorpus (11K sentences, 220K words).

5 Experimental Results

5.1 Parameter Optimization

The CLM has one free parameter, namey λ (Eq. (3)), the interpolation ratio between the document models and the collection model. Before

λ	0.1	0.2	0.3	0.4	0.5
$P_{\text{CLM}(s_n\|\emptyset)}$	149.6	**147.9**	150.2	155.9	165.0

Table 1: Parameter optimization: Per-word perplexity on the validation set for various values of λ.

	Model (with $\lambda = 0.2$)	Perplexity (% gain over $P_{\text{CLM}}(s_n\|\emptyset)$)
BLs	P_{M_C} (collection model)	154.429
	$P_{\text{CLM}}(s_n\|\emptyset)$	135.453
Experiment	$P_{\text{CLM}}(s_n\|s_{n-1})$	125.330 (7.47%)
	$P_{\text{CLM}}(s_n\|s_{n-2}s_{n-1})$	125.214 (7.55%)
	$P_{\text{CLM}}(s_n\|s_{n-3}\dots s_{n-1})$	124.098 (8.38%)
	$P_{\text{CLM}}(s_n\|s_{n-4}\dots s_{n-1})$	126.750 (6.42%)
	$P_{\text{CLM}}(s_n\|s_1\dots s_{n-1})$	130.426 (3.71%)
	$P_{\text{CLM}}(s_n\|s_{\text{title}})$	130.734 (3.48%)
UBs	$P_{\text{CLM}}(s_n\|s_{n-1}s_n)$	93.496 (30.97%)
	$P_{\text{CLM}}(s_n\|s_{n-2}\dots s_n)$	100.722 (25.64%)
	$P_{\text{CLM}}(s_n\|s_{n-3}\dots s_n)$	106.559 (21.33%)

Table 2: Per-word perplexity (sentences as targets): baselines (BLs), experiment, upper bounds (UBs)

proceeding to the final evaluation, we optimize λ on our validation set. Since we assume that the document models are fairly sparse and high values of λ correspond to document model dominance, we only consider λ values between 0.1 and 0.5.

Table 1 shows the perplexities of the baseline CLM model without context ($P_{\text{CLM}}(s_n|\emptyset)$) for various values of λ. The selection of λ heavily affects the model, with generally better perplexity for lower values of λ. This matches our intuition: we need to strongly smooth the document models with the collection model. However, the document models are informative after all. We achieve the highest reduction in perplexity for $\lambda = 0.2$. We use this value for the remainder of the experiments.

5.2 Main Evaluation

The results of our main experiment are shown in Table 2, which consists of three parts. The top part of Table 2 shows the two baselines. Note that the CLM without context ($P_{\text{CLM}}(s_n|\emptyset)$) already performs substantially better than the collection model P_{M_C}. $P_{\text{CLM}}(s_n|\emptyset)$ is essentially the average probability of generating s_n in the query likelihood model. It already takes benefit of document-level statistics in addition to collection-level statistics, which results in better estimation. Correspondingly, we adopt $P_{\text{CLM}}(s_n|\emptyset)$ as point of reference for all comparisons concerning the effect of context. All gains reported in the table are relative to this model.

The middle part of Table 2 shows the results for various settings of the Conditioned Language Model. We estimate the probability of individual target sentences, comparing various definitions of context as conditioning events as to their effectiveness in predicting the target. More specifically, we consider sentence windows of one to four previous sentences before the target text as well as longer discourse context, such as all preceding sentences in the document or the document tiele. For example, for each sentence s_n, the model $P(s_n|s_{n-1}s_{n-2})$ uses the two previous sentences, s_{n-1} and s_{n-2}, together as context.

All CLM models with context improve over the base model $P_{\text{CLM}}(s_n|\emptyset)$. Significance testing with bootstrap resampling (Efron and Tibshirani, 1993) showed that all performance gains are significant (all models: $p<0.001$). The best context among the evaluated models is a three-sentence window before the target sentence, which reduces the per-word perplexity by 8.38% compared to the null context CLM, a reduction of 19.64% compared to the collection model P_{M_C}. Both two-sentence and four-sentence models do clearly worse. It appears that the three-sentence window strikes the best balance between providing a rich context and diluting the local information too much. In comparison, wider discourse context performs much worse: the two CLM versions that take the complete prior context or the document title into account only obtain complexity reductions of between 3% and 4%. Our interpretation is that the CLM is able to pick up a modest amount of discourse coherence in terms of lexical distributions that slowly changes over the course of a document.

The bottom part of Table 2 aims at establishing an upper bound for the perplexity improvements that can be expected from the CLM by including the target sentence into the context. For example, the model $P_{\text{CLM}}(s_n|s_{n-1}s_n)$ uses the target sentence itself and its previous sentence as the context. Our rationale comes from the application of the CLM to tasks like sentence completion (Section 2.2). This involves a research question in its own right, namely defining which part of the problems should serve as the context and which as the text. While the split

can simply be made along phrase boundaries (P(broke my hand | I fell from a ladder and)), we believe that better results can be obtained if some parts of the problem are included in both t and c. For example, P(broke my hand | I fell from a ladder and broke) asks the model simultaneously to focus on documents that talk both about ladders and about breaking. In general, it seems a good idea to make as rich as possible both the context (for good document selection) and the text (for good plausibility estimation). Our "upper bound" models show the limit of this approach when the text is a proper subset of the context.

The results show that in this setup, sentence-window CLMs reduce perplexity greatly. The best model does so by 30.97%. It is the one-sentence window CLM, which is expected since larger contexts dilute the target sentence information.

6 Related Work

In n-gram LMs, more context can be integrated by simply increasing n. While the resulting complexity and efficiency issues can be addressed (Talbot and Osborne, 2007; Wood et al., 2009), it is difficult to go beyond n=5 even with trillions of words (Brants and Franz, 2006).

The CLM can be regarded as a type of adaptive LM. Adaptive LMs generally construct full-fledged models from specific datasets such as domains (Rosenfeld, 1996; Lin et al., 2011; Shi et al., 2012), LDA-style topics (Hsu and Glass, 2006; Trnka, 2008), or occurrences of individual words (Sicilia-Garcia et al., 2000). Once generated, the models are combined based on their match with the test topic or domain. Among such domain adaptation approaches, *training data selection* (Moore and Lewis, 2010; Axelrod et al., 2011) is most related to our work. It focuses on a small part of the training corpus particularly similar to the test domain. This is mirrored in the CLM's use of $P(d_i|c)$ to weigh documents based on context.

The two main differences are: (1), the selection is not made on the basis of a corpus, but of a relatively small context; (2), our CLM is more dynamic: the weighting is not given at training time, but by specifying a context at test time.

Other previous studies have explicitly introduced novel modeling strategies to incorporate long distance dependencies such as caching (Iyer and Ostendorf, 1999), triggering events (Rosenfeld, 1996), or neural networks (Schwenk, 2007; Mikolov and Zweig, 2012). Compared to these approaches, the CLM has two advantages: (a) it can be seen as a wrapper around standard LMs and can thus take advantage of all previous research; (b) it supports a wide range of context definitions, while previous work hard-coded context types.

7 Conclusion

This paper presents the Conditioned Language Model, a meta language model which can incorporate discourse context or previous knowledge. It models P(text|context), where both text and context can be arbitrary word sequences. We have described an approximation to make computation feasible for large document collections, and our preliminary evaluation shows that a small window context helps predicting target sentences, reducing per-word perplexity by 8.4% compared to the model without context. We interpret this as encouragement that the CLM can providing judgments about the likelihood of texts that incorporate discourse information in a natural and general manner, going beyond the capabilities of traditional n-gram LMs.

Our next steps will address more thorough evaluation of the model. It can replace LMs used in applications like MT or ASR. However, what we feel to be more promising is the use of CLM's conditional probabilities for "semantic" NLP tasks such as lexical substitution or cloze completion (cf. Section 2.2). Much work on such tasks is based on lexical association measures at the word level such as pointwise mutual information. The CLM can be understood as a natural generalization, namely an association measure at the sentence level, based on document distributions.

Acknowledgments.

We gratefully acknowledge partial funding by the European Commission (project EXCITEMENT (FP7 ICT-287923)).

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain

data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, United Kingdom.

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram version 1. Technical report, Linguistic Data Consortium, Philadelphia.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, New York.

William A Gale and Geoffrey Sampson. 1995. Good-turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237.

Bo-June (Paul) Hsu and James Glass. 2006. Style & topic language model adaptation using hmm-lda. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 373–381, Sydney, Australia.

Rukmini M. Iyer and Mari Ostendorf. 1999. Modeling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39.

Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Jimmy Lin, Rion Snow, and William Morgan. 2011. Smoothing techniques for adaptive online language models: Topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 422–429, San Diego, CA.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

Diana McCarthy and Roberto Navigli. 2009. The english lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *Proceedings of the IEEE Spoken Language Technology Workshop*, pages 234–239, Miami, FL, USA.

David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214–221, Berkeley, CA.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition. Technical report, Linguistic Data Consortium, Philadelphia.

Roberto Pieraccini. 2012. *The Voice in the Machine: Building Computers That Understand Speech*. The MIT Press.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281.

Ronald Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech & Language*, 10(3):187–228.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21(3):492–518.

Yangyang Shi, Pascal Wiggers, and Catholijn M Jonker. 2012. Adaptive language modeling with a set of domain dependent models. In *Proceedings of Text, Speech and Dialogue*, pages 472–479, Brno, Czech Republic.

E. I. Sicilia-Garcia, Ji Ming, and F. J. Smith. 2000. A dynamic language model based on individual word domains. In *Proceedings of the 18th Conference on Computational Linguistics*, pages 789–794, Saarbrücken, Germany.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of 7th International Conference on Spoken Language Processing*, pages 1045–1048, Denver, CO.

David Talbot and Miles Osborne. 2007. Smoothed bloom filter language models: Tera-scale LMs on the cheap. In *Proceedings of EMNLP*, pages 468–476, Prague, Czech Republic.

Keith Trnka. 2008. Adaptive language modeling for word prediction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*, pages 61–66, Columbus, Ohio.

Frank Wood, Cédric Archambeau, Jan Gasthaus, Lancelot James, and Yee Whye Teh. 2009. A stochastic memoizer for sequence data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1129–1136, Montreal, Canada.

Geoffrey Zweig and Chris J. C. Burges. 2012. A challenge set for advancing language modeling. In *Proceedings of the NAACL-HLT 2012 Workshop on the Future of Language Modeling for HLT*, pages 29–36, Montreal, Canada.