Improving Text Simplification Language Modeling Using Unsimplified Text Data

David Kauchak
Middlebury College
Middlebury, VT 05753
dkauchak@middlebury.edu

Abstract

In this paper we examine language modeling for text simplification. Unlike some text-to-text translation tasks, text simplification is a monolingual translation task allowing for text in both the input and output domain to be used for training the language model. We explore the relationship between normal English and simplified English and compare language models trained on varying amounts of text from each. We evaluate the models intrinsically with perplexity and extrinsically on the lexical simplification task from SemEval 2012. We find that a combined model using both simplified and normal English data achieves a 23% improvement in perplexity and a 24% improvement on the lexical simplification task over a model trained only on simple data. Post-hoc analysis shows that the additional unsimplified data provides better coverage for unseen and rare n-grams.

1 Introduction

An important component of many text-to-text translation systems is the language model which predicts the likelihood of a text sequence being produced in the output language. In some problem domains, such as machine translation, the translation is between two distinct languages and the language model can only be trained on data in the output language. However, some problem domains (e.g. text compression, text simplification and summarization) can be viewed as monolingual translation tasks, translating between text variations within a single language. In these monolingual problems, text *could* be used from both the input and output domain to train a language model. In this paper, we investigate this possibility for text simplification where both simplified English text and normal English text are available for training a simple English language model.

Table 1 shows the n-gram overlap proportions in a sentence aligned data set of 137K sentence pairs from aligning Simple English Wikipedia and English Wikipedia articles (Coster and Kauchak, 2011a).[1] The data highlights two conflicting views: does the benefit of additional data outweigh the problem of the source of the data? Throughout the rest of this paper we refer to sentences/articles/text from English Wikipedia as *normal* and sentences/articles/text from Simple English Wikipedia as *simple*.

On the one hand, there is a strong correspondence between the simple and normal data. At the word level 96% of the simple words are found in the normal corpus and even for n-grams as large as 5, more than half of the n-grams can be found in the normal text. In addition, the normal text does represent English text and contains many n-grams not seen in the simple corpus. This extra information may help with data sparsity, providing better estimates for rare and unseen n-grams.

On the other hand, there is still only modest overlap between the sentences for longer n-grams, particularly given that the corpus is sentence-aligned and that 27% of the sentence pairs in this aligned data set are identical. If the word distributions were very similar between simple and normal text, then the overlap proportions between the two languages would be similar regardless of which direction the comparison is made. Instead, we see that the normal text has more varied language and contains more n-grams. Previous research has also shown other differences between simple and normal data sources that could impact language model performance including average number of syllables, reading

[1] http://www.cs.middlebury.edu/~dkauchak/simplification

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1537–1546,
Sofia, Bulgaria, August 4-9 2013.

n-gram size:	1	2	3	4	5
simple in normal	0.96	0.80	0.68	0.61	0.55
normal in simple	0.87	0.68	0.58	0.51	0.46

Table 1: The proportion of n-grams that overlap in a corpus of 137K sentence-aligned pairs from Simple English Wikipedia and English Wikipedia.

complexity, and grammatical complexity (Napoles and Dredze, 2010; Zhu et al., 2010; Coster and Kauchak, 2011b). In addition, for some monolingual translation domains, it has been argued that it is not appropriate to train a language model using data from the input domain (Turner and Charniak, 2005).

Although this question arises in other monolingual translation domains, text simplification represents an ideal problem area for analysis. First, simplified text data is available in reasonable quantities. Simple English Wikipedia contains more than 60K articles written in simplified English. This is not the case for all monolingual translation tasks (Knight and Marcu, 2002; Cohn and Lapata, 2009). Second, the quantity of simple text data available is still limited. After preprocessing, the 60K articles represents less than half a million sentences which is orders of magnitude smaller than the amount of normal English data available (for example the English Gigaword corpus (David Graff, 2003)). Finally, many recent text simplification systems have utilized language models trained only on simplified data (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011a; Wubben et al., 2012); improvements in simple language modeling could translate into improvements for these systems.

2 Related Work

If we view the normal data as out-of-domain data, then the problem of combining simple and normal data is similar to the language model domain adaption problem (Suzuki and Gao, 2005), in particular *cross-domain adaptation* (Bellegarda, 2004) where a domain-specific model is improved by incorporating additional general data. Adaptation techniques have been shown to improve language modeling performance based on perplexity (Rosenfeld, 1996) and in application areas such as speech transcription (Bacchiani and Roark, 2003) and machine translation (Zhao et al., 2004), though no previous research has examined the language model domain adaptation problem for text simplification. Pan and Yang (2010) provide a survey on the related problem of domain adaptation for machine learning (also referred to as "transfer learning"), which utilizes similar techniques. In this paper, we explore some basic adaptation techniques, however this paper is not a comparison of domain adaptation techniques for language modeling. Our goal is more general: to examine the relationship between simple and normal data and determine whether normal data is helpful. Previous domain adaptation research is complementary to our experiments and could be explored in the future for additional performance improvements.

Simple language models play a role in a variety of text simplification applications. Many recent statistical simplification techniques build upon models from machine translation and utilize a simple language model during simplification/decoding both in English (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011a; Wubben et al., 2012) and in other languages (Specia, 2010). Simple English language models have also been used as predictive features in other simplification sub-problems such as lexical simplification (Specia et al., 2012) and predicting text simplicity (Eickhoff et al., 2010).

Due to data scarcity, little research has been done on language modeling in other monolingual translation domains. For text compression, most systems are trained on uncompressed data since the largest text compression data sets contain only a few thousand sentences (Knight and Marcu, 2002; Galley and McKeown, 2007; Cohn and Lapata, 2009; Nomoto, 2009). Similarly for summarization, systems that have employed language models trained only on unsummarized text (Banko et al., 2000; Daume and Marcu, 2002).

3 Corpus

We collected a data set from English Wikipedia and Simple English Wikipedia with the former representing normal English and the latter simple English. Simple English Wikipedia has been previously used for many text simplification approaches (Zhu et al., 2010; Yatskar et al., 2010; Biran et al., 2011; Coster and Kauchak, 2011a; Woodsend and Lapata, 2011; Wubben et al., 2012) and has been shown to be simpler than normal English Wikipedia by both automatic measures and human perception (Coster and Kauchak, 2011b;

	simple	normal
sentences	385K	2540K
words	7.15M	64.7M
vocab size	78K	307K

Table 2: Summary counts for the simple-normal article aligned data set consisting of 60K article pairs.

Woodsend and Lapata, 2011). We downloaded **all** articles from Simple English Wikipedia then removed stubs, navigation pages and any article that consisted of a single sentence, resulting in 60K simple articles.

To partially normalize for content and source differences we generated a document aligned corpus for our experiments. We extracted the corresponding 60K normal articles from English Wikipedia based on the article title to represent the normal data. We held out 2K article pairs for use as a testing set in our experiments. The extracted data set is available for download online.[2]

Table 2 shows count statistics for the collected data set. Although the simple and normal data contain the same number of articles, because normal articles tend to be longer and contain more content, the normal side is an order of magnitude larger.

4 Language Model Evaluation: Perplexity

To analyze the impact of data source on simple English language modeling, we trained language models on varying amounts of simple data, normal data, and a combination of the two. For our first task, we evaluated these language models using perplexity based on how well they modeled the simple side of the held-out data.

4.1 Experimental Setup

We used trigram language models with interpolated Kneser-Kney discounting trained using the SRI language modeling toolkit (Stolcke, 2002). To ensure comparability, all models were closed vocabulary with the same vocabulary set based on the words that occurred in the simple side of the training corpus, though similar results were seen for other vocabulary choices. We generated different models by varying the size and type of training data:

- **simple-only**: simple sentences only
- **normal-only**: normal sentences only
- **simple-*X*+normal**: *X* simple sentences combined with a varying number of normal sentences

To evaluate the language models we calculated the model *perplexity* (Chen et al., 1998) on the simple side of the held-out data. The test set consisted of 2K simple English articles with 7,799 simple sentences and 179K words. Perplexity measures how likely a model finds a test set, with lower scores indicating better performance.

[2] http://www.cs.middlebury.edu/~dkauchak/simplification

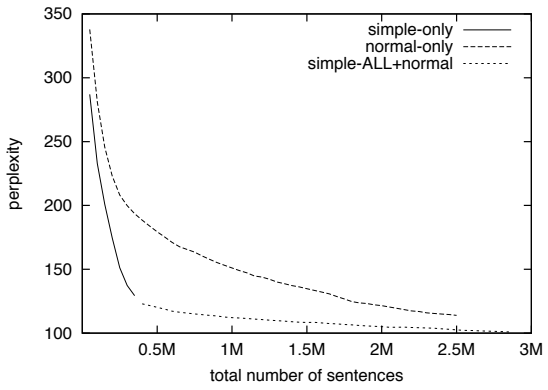

Figure 1: Language model perplexities on the held-out test data for models trained on increasing amounts of data.

4.2 Perplexity Results

Figure 1 shows the language model perplexities for the three types of models for increasing amounts of training data. As expected, when trained on the same amount of data, the language models trained on simple data perform significantly better than language models trained on normal data. In addition, as we increase the amount of data, the simple-only model improves more than the normal-only model.

However, the results also show that the normal data does have some benefit. The perplexity for the *simple-ALL+normal* model, which starts with *all* available simple data, continues to improve as normal data is added resulting in a 23% improvement over the model trained with only simple data (from a perplexity of 129 down to 100). Even by itself the normal data does have value. The normal-only model achieves a slightly better perplexity than the simple-only model, though only by utilizing an order of magnitude more data.

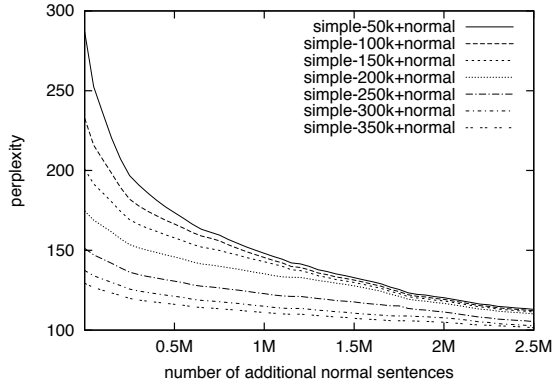

Figure 2: Language model perplexities for combined simple-normal models. Each line represents a model trained on a different amount of simple data as normal data is added.

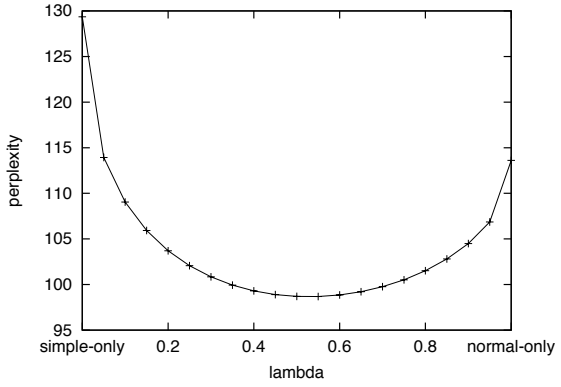

Figure 3: Perplexity scores for a linearly interpolated model between the simple-only model and the normal-only model for varying lambda values.

To better understand how the amount of simple and normal data impacts perplexity, Figure 2 shows perplexity scores for models trained on varying amounts of simple data as we add increasing amounts of normal data. We again see that normal data is beneficial; regardless of the amount of simple data, adding normal data improves perplexity. This improvement is most beneficial when simple data is limited. Models trained on less simple data achieved larger performance increases than those models trained on more simple data.

Figure 2 also shows again that simple data is more valuable than normal data. For example, the simple-only model trained on 250K sentences achieves a perplexity of approximately 150. To achieve this same perplexity level starting with 200K simple sentences requires an additional 300K normal sentences, or starting with 100K simple sentences an additional 850K normal sentences.

4.3 Language Model Adaptation

In the experiments above, we generated the language models by treating the simple and normal data as one combined corpus. This approach has the benefit of simplicity, however, better performance for combining related corpora has been seen by domain adaptation techniques which combine the data in more structured ways (Bacchiani and Roark, 2003). *Our goal for this paper is not to explore domain adaptation techniques, but to determine if normal data is useful for the simple language modeling task.* However, to provide another dimension for comparison and to understand if domain adaptation techniques may be useful, we also investigated a linearly interpolated language model.

A linearly interpolated language model combines the probabilities of two or more language models as a weighted sum. In our case, the interpolated model combines the simple model estimate, $p_s(w_i|w_{i-2}, w_{i-1})$, and the normal model estimate, $p_n(w_i|w_{i-2}, w_{i-1})$, linearly (Jelinek and Mercer, 1980; Hsu, 2007):

$$p_{interpolated}(w_i|w_{i-2}, w_{i-1}) = \lambda\, p_n(w_i|w_{i-2}, w_{i-1}) + (1-\lambda)\, p_s(w_i|w_{i-2}, w_{i-1})$$

where $0 \geq \lambda \geq 1$.

Figure 3 shows perplexity scores for varying lambda values ranging from the simple-only model on the left with $\lambda = 0$ to the normal-only model on the right with $\lambda = 1$. As with the previous experiments, adding normal data improves improves perplexity. In fact, with a lambda of 0.5 (equal weight between the models) the performance is slightly better than the aggregate approaches above with a perplexity of 98. The results also highlight the balance between simple and normal data; normal data is not as good as simple data and adding too much of it can cause the results to degrade.

5 Language Model Evaluation: Lexical Simplification

Currently, no automated methods exist for evaluating sentence-level or document-level text simplification systems and manual evaluation is time-consuming, expensive and has not been validated. Because of these evaluation challenges, we chose to evaluate the language models extrinsi-

Word:	*tight*
Context:	With the physical market as **tight** as it has been in memory, silver could fly at any time.
Candidates:	constricted, pressurised, low, high-strung, tight
Human ranking:	tight, low, constricted, pressurised, high-strung

Figure 4: A lexical substitution example from the SemEval 2012 data set.

cally based on the lexical simplification task from SemEval 2012 (Specia et al., 2012).

Lexical simplification is a sub-problem of the general text simplification problem (Chandrasekar and Srinivas, 1997); a sentence is simplified by substituting words or phrases in the sentence with "simpler" variations. Lexical simplification approaches have been shown to improve the readability of texts (Urano, 2000; Leroy et al., 2012), are useful in domains such as medical texts where major content changes are restricted, and they may be useful as a pre- or post-processing step for general simplification systems.

5.1 Experimental Setup

Examples from the lexical simplification data set from SemEval 2012 consist of three parts: w, the word to be simplified; $s_1, ..., s_{i-1}, w, s_{i+1}, ..., s_n$, a sentence containing the word; and, $r_1, r_2, ..., r_m$, a list of candidate simplifications for w. The goal of the task is to rank the candidate simplifications according to their simplicity in the context of the sentence. Figure 4 shows an example from the data set. The data set contains a development set of 300 examples and a test set of 1710 examples.[3] For our experiments, we evaluated the models on the test set.

Given a language model $p(\cdot)$ and a lexical simplification example, we ranked the list of candidates based on the probability the language model assigns to the sentence with the candidate simplification inserted in context. Specifically, we scored each candidate simplification r_j by

$p(s_1 ... \ s_{i-1} \ r_j \ s_{i+1} ... \ s_n)$

and then ranked them based on this score. For example, to calculate the ranking for the example in Figure 4 we calculate the probability of each of:

With the physical market as *constricted* as it has been ...
With the physical market as *pressurised* as it has been ...
With the physical market as *low* as it has been ...
With the physical market as *high-strung* as it has been ...
With the physical market as *tight* as it has been ...

with the language model and then rank them by their probability. *We do not suggest this as a complete lexical substitution system*, but it was a common feature for many of the submitted systems, it performs well relative to the other systems, and it allows for a concrete comparison between the language models on a simplification task.

[3] http://www.cs.york.ac.uk/semeval-2012/task1/

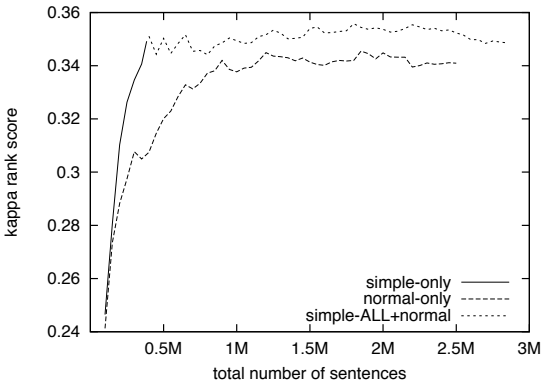

Figure 5: Kappa rank scores for the models trained on increasing amounts of data.

To evaluate the rankings, we use the metric from the SemEval 2012 task, the Cohen's kappa coefficient (Landis and Koch, 1977) between the system ranking and the human ranking, which we denote the "kappa rank score". See Specia et al. (2012) for the full details of how the evaluation metric is calculated.

We use the same setup for training the language models as in the perplexity experiments except the models are open vocabulary instead of closed. Open vocabulary models allow for the language models to better utilize the varying amounts of data and since the lexical simplification problem only requires a comparison of probabilities within a given model to produce the final ranking, we do not need the closed vocabulary requirement.

5.2 Lexical Simplification Results

Figure 5 shows the kappa rank scores for the simple-only, normal-only and combined models. As with the perplexity results, for similar amounts of data the simple-only model performs better than the normal-only model. We also again see that the performance difference between the two models grows as the amount of data increases. However,

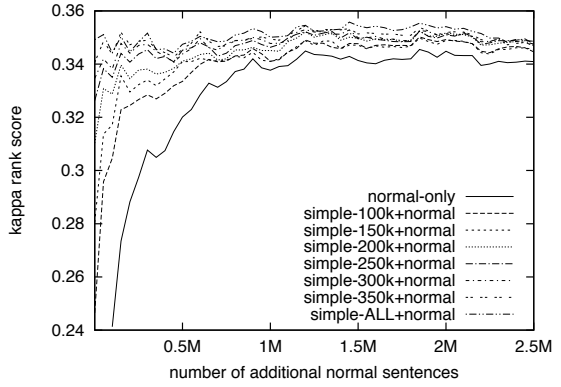

Figure 6: Kappa rank scores for models trained with varying amounts of simple data combined with increasing amounts of normal data.

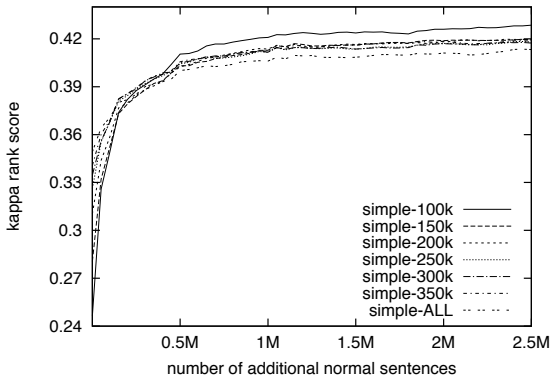

Figure 7: Kappa rank scores for linearly interpolated models between simple-only and normal-only models trained with varying amounts of simple and normal data.

unlike the perplexity results, simply appending additional normal data to the entire simple data set does not improve the performance of the lexical simplifier.

To determine if additional normal data improves the performance for models trained on smaller amounts of simple data, Figure 6 shows the kappa rank scores for models trained on different amounts of simple data as additional normal data is added. For smaller amounts of simple data adding normal data does improve the kappa rank score. For example, a language model trained with 100K simple sentences achieves a score of 0.246 and is improved by almost 40% to 0.344 by adding all of the additional normal data. Even the performance of a model trained with 300K simple sentences is increased by 3% (0.01 improvement in kappa rank score) by adding normal data.

5.3 Language Model Adaptation

The results in the previous section show that adding normal data to a simple data set can improve the lexical simplifier if the amount of simple data is limited. To investigate this benefit further, we again generated linearly interpolated language models between the simple-only model and the normal-only model. Figure 7 shows results for the same experimental design as Figure 6 with varying amounts of simple and normal data, however, rather than appending the normal data we trained the models separately and created a linearly interpolated model as described in Section 4.3. The best lambda was chosen based on a linear search optimized on the SemEval 2012 development set.

For all starting amounts of simple data, interpolating the simple model with the normal model results in a large increase in the kappa rank score. *Combining the model trained on all the simple data with the model trained on all the normal data achieves a score of 0.419, an improvement of 23% over the model trained on only simple data.* Although our goal was not to create the best lexical simplification system, this approach would have ranked 6th out of 11 submitted systems in the SemEval 2012 competition (Specia et al., 2012).

Interestingly, although the performance of the simple-only models varied based on the amount of simple data, when these models are interpolated with a model trained on normal data, the performance tended to converge. This behavior is also seen in Figure 6, though to a lesser extent. This may indicate that for some tasks like lexical simplification, only a modest amount of simple data is required when combining with additional normal data to achieve reasonable performance.

6 Why Does Unsimplified Data Help?

For both the perplexity experiments and the lexical simplification experiments, utilizing additional normal data resulted in large performance improvements; using **all** of the simple data available, performance is still significantly improved when combined with normal data. In this section, we investigate why the additional normal data is beneficial for simple language modeling.

6.1 More n-grams

Intuitively, adding normal data provides additional English data to train on. Most language models are

	Perplexity test data			Lexical simplification		
	simple	normal	% inc.	simple	normal	% inc.
1-grams	0.85	0.93	9.4%	0.74	0.78	6.2%
2-grams	0.66	0.82	24%	0.34	0.54	56%
3-grams	0.39	0.57	46%	0.088	0.19	117%

Table 3: Proportion of n-grams in the test sets that occur in the simple and normal training data sets.

	Perplexity test data		Lexical simplification	
	simple + normal	% inc. over normal	simple + normal	% inc. over normal
1-grams	0.93	0.2%	0.78	0.0%
2-grams	0.83	0.8%	0.54	1.1%
3-grams	0.58	2.5%	0.20	2.6%

Table 4: Proportion of n-grams in the test sets that occur in the *combination* of both the simple and normal data.

trained using a smoothed version of the maximum likelihood estimate for an n-gram. For trigrams, this is:

$$p(a|bc) = \frac{count(abc)}{count(bc)}$$

where $count(\cdot)$ is the number of times the n-gram occurs in the training corpus. For interpolated and backoff n-gram models, these counts are smoothed based on the probabilities of lower order n-gram models, which are in-turn calculated based on counts from the corpus.

We hypothesize that the key benefit of additional normal data is access to more n-gram counts and therefore better probability estimation, particularly for n-grams in the simple corpus that are unseen or have low frequency. For n-grams that have never been seen before, the normal data provides some estimate from English text. This is particularly important for unigrams (i.e. words) since there is no lower order model to gain information from and most language models assume a uniform prior on unseen words, treating them all equally. For n-grams that have been seen but are rare, the additional normal data can help provide better probability estimates. Because frequencies tend to follow a Zipfian distribution, these rare n-grams make up a large portion of n-grams in real data (Ha et al., 2003).

To partially validate this hypothesis, we examined the n-gram overlap between the n-grams in the training data and the n-grams in the test sets from the two tasks. Table 3 shows the percentage of unigrams, bigrams and trigrams from the two test sets that are found in the simple and normal training data.

For all n-gram sizes the normal data contained more test set n-grams than the simple data. Even at the unigram level, the normal data contained significantly more of the test set unigrams than the simple data. On the perplexity data set, the 9.4% increase in word occurrence between the simple and normal data set represents an over 50% reduction in the number of out of vocabulary words. For larger n-grams, the difference between the simple and normal data sets are even more pronounced. On the lexical simplification data the normal data contained more than twice as many test trigrams as the simple data. These additional n-grams allow for better probability estimates on the test data and therefore better performance on the two tasks.

6.2 The Role of Normal Data

Estimation of rare events is one component of language model performance, but other factors also impact performance. Table 4 shows the test set n-gram overlap on the combined data set of simple and normal data. Because the simple and normal data come from the same content areas, the simple data provides little additional coverage if the normal data is already used. For example, adding the simple data to the normal data only increases the number of seen unigrams by 0.2%, representing only about 600 new words. However, the experiments above showed the combined models performed much better than models trained only on normal data.

This discrepancy highlights the key problem with normal data: it is out-of-domain data. While it shares some characteristics with the simple data, it represents a different distribution over the language. To make this discrepancy more explicit, we created a sentence aligned data set by aligning the simple and normal articles using the approach from Coster and Kauchak (2011b). This approach has been previously used for aligning English Wikipedia and Simple English Wikipedia with reasonable accuracy. The resulting data set contains 150K aligned simple-normal sentence pairs.

Figure 8 shows the perplexity scores for language models trained on this data set. Because the data is aligned and therefore similar, we see the perplexity curves run parallel to each other as more data is added. However, even though these

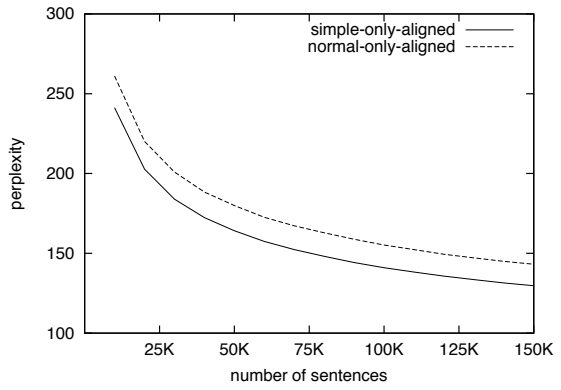

Figure 8: Language model perplexities for models trained on increasing data sizes for a simple-normal sentence aligned data set.

sentences represent the same content, the language use is different between simple and normal and the normal data performs consistently worse.

6.3 A Balance Between Simple and Normal

Examining the optimal lambda values for the linearly interpolated models also helps understand the role of the normal data. On the perplexity task, the best perplexity results were obtained with a lambda of 0.5, or an equal weighting between the simple and normal models. Even though the normal data contained six times as many sentences and nine times as many words, the best modeling performance balanced the quality of the simple model with the coverage of the normal model.

For the simplification task, the optimal lambda value determined on the development set was 0.98, with a very strong bias towards the simple model. Only when the simple model did not provide differentiation between lexical choices will the normal model play a role in selecting the candidates. For the lexical simplification task, the role of the normal model is even more clear: to handle rare occurrences not covered by the simple model and to smooth the simple model estimates.

7 Conclusions and Future Work

In the experiments above we have shown that on two different tasks utilizing additional normal data improves the performance of simple English language models. On the perplexity task, the combined model achieved a performance improvement of 23% over the simple-only model and on the lexical simplification task, the combined model achieved a 24% improvement. These improvements are achieved over a simple-only model that uses *all* simple English data currently available in this domain.

For both tasks, the best improvements were seen when using language model adaptation techniques, however, the adaptation results also indicated that the role of normal data is partially task dependent. On the perplexity task, the best results were achieved with an equal weighting between the simple-only and normal-only model. However, on the lexical simplification task, the best results were achieved with a very strong bias towards the simple-only model. For other simplification tasks, the optimal parameters will need to be investigated.

For many of the experiments, combining a smaller amount of simple data (50K-100K sentences) with normal data achieved results that were similar to larger simple data set sizes. For example, on the lexical simplification task, when using a linearly interpolated model, the model combining 100K simple sentences with all the normal data achieved comparable results to the model combining all the simple sentences with all the normal data. This is encouraging for other monolingual domains such as text compression or text simplification in non-English languages where less data is available.

There are still a number of open research questions related to simple language modeling. First, further experiments with larger normal data sets are required to understand the limits of adding out-of-domain data. Second, we have only utilized data from Wikipedia for normal text. Many other text sources are available and the impact of not only size, but also of domain should be investigated. Third, it still needs to be determined how language model performance will impact sentence-level and document-level simplification approaches. In machine translation, improved language models have resulted in significant improvements in translation performance (Brants et al., 2007). Finally, in this paper we only investigated linearly interpolated language models. Many other domain adaptations techniques exist and may produce language models with better performance.

References

Michiel Bacchiani and Brian Roark. 2003. Unsupervised language model adaptation. In *Proceedings of ICASSP*.

Michele Banko, Vibhu Mittal, and Michael Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of ACL*.

Jerome R. Bellegarda. 2004. Statistical language model adaptation: Review and perspectives. *Speech Communication*.

Or Biran, Samuel Brody, and Noemie Elhadad. 2011. Putting it simply: A context-aware approach to lexical simplification. In *Proceedings of ACL*.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of EMNLP*.

Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge Based Systems*.

Stanley Chen, Douglas Beeferman, and Ronald Rosenfeld. 1998. Evaluation metrics for language models. In *DARPA Broadcast News Transcription and Understanding Workshop*.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*.

William Coster and David Kauchak. 2011a. Learning to simplify sentences using Wikipedia. In *Proceedings of Text-To-Text Generation*.

William Coster and David Kauchak. 2011b. Simple English Wikipedia: A new text simplification task. In *Proceedings of ACL*.

Hal Daume and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of ACL*.

Christopher Cieri David Graff. 2003. English gigaword. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05`.

Carsten Eickhoff, Pavel Serdyukov, and Arjen P. de Vries. 2010. Web page classification on child suitability. In *Proceedings of CIKM*.

Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *Proceedings of HLT-NAACL*.

Le Quan Ha, E. I. Sicilia-Garcia, Ji Ming, and F. J. Smith. 2003. Extension of Zipf's law to word and character n-grams for English and Chinese. *Computational Linguistics and Chinese Language Processing*.

Bo-June Hsu. 2007. Generalized linear interpolation of language models. In *IEEE Workshop on ASRU*.

Frederick Jelinek and Robert Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Patter Recognition in Practice*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*.

Gondy Leroy, James E. Endicott, Obay Mouradi, David Kauchak, and Melissa Just. 2012. Improving perceived and actual text difficulty for health information consumers using semi-automated methods. In *American Medical Informatics Association (AMIA) Fall Symposium*.

Courtney Napoles and Mark Dredze. 2010. Learning simple Wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of HLT/NAACL Workshop on Computation Linguistics and Writing*.

Tadashi Nomoto. 2009. A comparison of model free versus model intensive approaches to sentence compression. In *Proceedings of EMNLP*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*.

Ronald Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *Joint Conference on Lexical and Computerational Semantics (*SEM)*.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *Proceedings of Computational Processing of the Portuguese Language*.

Andreas Stolcke. 2002. SRILM - An extensible language modeling toolkit. In *Proceedings of ICSLP*.

Hisami Suzuki and Jianfeng Gao. 2005. A comparative study on language model adaptation techniques. In *Proceedings of EMNLP*.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of ACL*.

Ken Urano. 2000. Lexical simplification and elaboration: Sentence comprehension and incidental vocabulary acquisition. Master's thesis, University of Hawaii.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of EMNLP*.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of ACL*.

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Proceedings of NAACL*.

Bing Zhao, Matthias Eck, and Stephan Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of COLING*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of ICCL*.