Synthesising isiZulu-English code-switch bigrams using word embeddings

Ewald van der Westhuizen, Thomas Niesler

Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa

ewaldvdw@sun.ac.za, trn@sun.ac.za

Abstract

Code-switching is prevalent among South African speakers, and presents a challenge to automatic speech recognition systems. It is predominantly a spoken phenomenon, and generally does not occur in textual form. Therefore a particularly serious challenge is the extreme lack of training material for language modelling. We investigate the use of word embeddings to synthesise isiZulu-to-English code-switch bigrams with which to augment such sparse language model training data. A variety of word embeddings are trained on a monolingual English web text corpus, and subsequently queried to synthesise code-switch bigrams. Our evaluation is performed on language models trained on a new, although small, English-isiZulu code-switch corpus compiled from South African soap operas. This data is characterised by fast, spontaneously spoken speech containing frequent code-switching. We show that the augmentation of the training data with code-switched bigrams synthesised in this way leads to a reduction in perplexity.

Index Terms: code-switching, word vectors, word embeddings, Zulu, IsiZulu, spontaneous speech

1. Introduction

South Africa is a multilingual society and it is very common for speakers to use more than one language within the same sentence or conversation. This phenomenon is known as code-switching (CS) and is a topic of current research interest in the field of automatic speech recognition (ASR). Almost all ASR systems are optimised for monolingual settings, and the presence of code-switching degrades performance. Currently language modelling across language boundaries is a key challenge in the ASR of code-switched speech. We consider language modelling for an English-isiZulu code-switched speech corpus compiled from broadcast South African soap operas.

Very few speech corpora that include code-switching have been described in the ASR literature. Small corpora have been compiled for English-Spanish [1, 2], Cantonese-English [3, 4], Hindi-English [5] and for Sepedi-English [6]. However, the language pair English-Mandarin has received by far the most attention [7–14]. Approaches to code-switched language modelling include interpolating n-gram language models (LM) trained on monolingual data [13], n-grams trained on code-switched data [5, 7], class-based n-grams using additional features [4], recurrent neural networks [10], and combinations of approaches [11]. A particularly relevant recent study considered features for factored language models for Mandarin-English code-switched speech using the SEAME corpus [12].

Word embeddings (WE) encompass a set of techniques to represent words and word contexts in a real-valued multi-dimensional vector space. WE models have been applied to information retrieval and to word similarity and analogy evaluation. Broadly speaking, two approaches currently exist to obtain such models. The first is based on co-occurrence counts collected as a word-context matrix, followed by dimension reduction through matrix factorisation [15]. The second approach employs neural networks to optimise a training goal, such as maximising a similarity score for vectors representing word-context pairs in a corpus [15–17]. In our application, we query WE models to obtain words that are similar to those occurring in known code-switched bigrams. We show that LM perplexity is improved when LM training data is augmented with synthesised code-switched bigrams generated from word embedding models trained on out-of-domain web text. The process is straightforward and has the advantage of integrating easily into conventional language model training and decoding architectures.

2. Corpus description

South African soap operas present interesting examples of code-switching in their deliberate effort to accommodate a language-diverse viewer base. A transcribed corpus of such English-isiZulu code-switched speech was recently introduced [18]. In this work we employ a language-balanced subset of this original corpus, which is dominated by English. The development and test sets, which consist exclusively of code-switched utterances, have remained unchanged, however, permitting the direct comparison of results (see Table 1). The corpus comprises 1.55 hours monolingual English, 1.55 hours monolingual isiZulu and 2.35 hours English-isiZulu code-switched speech.

In our data, many English words are preceded by a Zulu prefix and these were annotated as code-switching. Although we focus on language modelling, we note that the speech is spontaneous and fast, with speech rates of 18.45 phones per second for isiZulu and 14.61 phones per second for English [18].

In comparison with the SEAME corpus used in [12], our corpus is very small (63 vs 5.45 hours). The severe sparsity of our data discourages the use of neural network based approaches, which are subject to specialisation and overfitting. Furthermore, we do not have the luxury of filtering training examples with occurrence thresholds, as was done in [12]. In our case such filtering catastrophically reduces the size of the available dataset, rendering it impractically small. Finally, Mandarin exhibits strongly analytic properties and grammatical inflection hardly ever occurs. In contrast, isiZulu exhibits properties of strongly synthetic languages, where agglutination and grammatical inflection is the norm. Therefore the composition of code-switching we observe in English-isiZulu differs from that observed in Mandarin-English and may require different methods.

3. Analysis

Our corpus contains 6816 bigrams that span a language switch. Table 2 shows the token counts for these CS bigrams for Zulu to English (ZEBG) and English to Zulu (EZBG). The distribution is fairly even between the training, development and test sets.

Table 1: *English-isiZulu code-switched (CS) corpus training, development and test sets.*

	Train	Dev	Test	Total
Duration	4.81h	8min	30.4min	5.45h
Word tokens	52.4k	1.6k	5.7k	59.6k
Word types	10.4k	858	2.3k	11.3k
Utterance count	8381	225	768	9374

Table 2: *Occurrence counts of CS bigram tokens (ZEBG: Zulu-to-English CS bigrams; EZBG: English-to-Zulu CS bigrams).*

	Train	Dev	Test	Total
ZEBG	2743	198	776	3717
EZBG	2236	175	688	3099

Table 3 shows the occurrence distribution of ZEBGs. ZEBGs occurring exactly once and exactly twice account for 75% and 11.88%, respectively, of all ZEBGs in the training set. Fewer than 13% occur 3 times or more, and none occur more than 26 times. This highlights the severe sparsity of the data.

Table 3: *Occurrence count distribution of ZEBG types in the training set and their percentage occupancy.*

#Typ	2064	163	38	14	9	6	2	2	1	1	1	1
Occ	1	2	3	4	5	6	9	13	7	10	15	26
%	75.24	11.88	4.15	2.04	1.64	1.31	.65	.94	.25	.36	.54	.94

Inspection of switch pairs in our corpus shows that many consist of a Zulu prefix followed by an English word. The prefix "zulufies" the English term. Consider the two example bigrams, *"i- album"* and *"ukuthi I'm"*. We will refer to the Zulu prefix *'i-'* as a trigger prefix, and the Zulu word *'ukuthi'* as a trigger word. Trigger prefixes account for just over half of all ZEBGs (51.11%), and trigger words for just under half (46.70%). The remaining 2.19% are cases in which a Zulu suffix is sandwiched between two English words. An example is *'finalise -anga like'*, where *'-anga'*, a past tense negative suffix, is linked to the verb *'finalise'*, but precedes *'like'* to form a switch bigram, *'-anga like'*, beginning with the suffix.

Table 4 indicates the seven most common Zulu trigger prefixes and words in the training set. Their classification and/or meanings are listed below [19]:

i- / ama- Classes 5 / 6 (singular / plural) noun prefixes

u- Class 1a (singular) noun prefix

ukuthi A conjunction similar to *'that'* or *'so that'*, and also the infinitive of *'to say'*.

e- Classes 4 / 9 (plural / singular) relative concord

ngi- Personal subject concord *'I'*, or personal object concord *'me'*.

le Classes 4 / 9 (plural / singular) demonstrative pronoun (*'this'*, *'that'*, *'that over there'*)

Grammatical noun classes, which are typical of Bantu languages, are analogous to grammatical gender found in many Indo-European languages. Each class groups related nouns together. For example, classes 1, 1a, 2 and 2a contain persons, while classes 9 and 10 contain animals. However, many exceptions exist for each class. Foreign words, such as English singular and plural nouns, typically fall in classes 5 and 6 respectively. This explains the high incidence of switches with *i-*.

Table 4: *The seven most common CS trigger types for ZEBGs in the train set (%ZEBG: percentage of the ZEBGs in train set).*

Type	*i-*	*u-*	*ukuthi*	*ama-*	*e-*	*ngi-*	*le*
Count	353	154	96	92	87	56	56
%ZEBG	12.7	5.61	3.49	3.35	3.17	2.04	2.04

Ukuthi, however, is often followed by personal pronouns.

Although Zulu trigger words make up about 47% of all ZEBGs, the word types are more diverse than the prefixes. The diversity of word types is in part attributed to agglutination. The number of word types grows quite rapidly, even for small data sets, and is also evident in other agglutinative languages [20]. Table 5 indicates trigger prefix types and trigger word types occurring once and twice among training set ZEBGs. We see that 18% of all trigger words occur only once. In contrast, only 4.15% of trigger prefix types occur only once. The subsequent experiments focus on the seven trigger types in Table 4.

Table 5: *Ratio of trigger types occurring exactly once and exactly twice among the code-switched bigrams (%ZEBG: percentage of the ZEBGs in training set).*

	Prefixes		Words	
Occurring:	**Count**	**%ZEBG**	**Count**	**%ZEBG**
Once	114	4.15	506	18.44
Twice	34	2.47	81	5.90

4. Experimental setup

Our aim is to predict probable Zulu-to-English bigrams and to use these to augment the sparse training data used for language modelling. We use the word similarities provided by word embeddings to synthesise Zulu-to-English bigrams. Figure 1 shows the steps involved in our experimental procedure, and the following sections discuss each step.

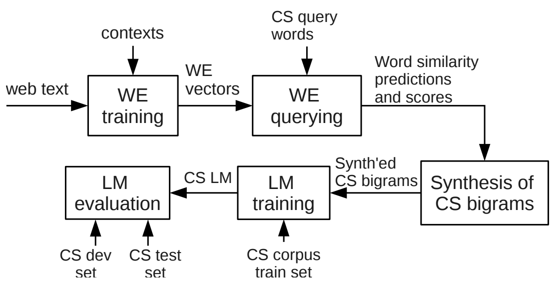

Figure 1: *Flow diagram depicting the steps followed in the experimental procedure.*

4.1. Training word embedding models

The open source tools, *hyperwords* [15], are used to train word embedding (WE) models on a monolingual South African English web text corpus consisting of 1 million sentences. The vocabulary of the web text is limited to the English vocabulary of the CS corpus. We use the matrix factorisation based *corpus2svd* training procedure. This involves the generation of a word-context co-occurrence matrix from the training data, followed by a pointwise mutual information (PMI) step and singular value decomposition (SVD) for dimension reduction. The training parameters optimised on the development set are:

- context window length, and
- dimension of output word vectors.

We use the toolkit default values for other training parameters.

The output of the WE training is a word vector model, which maps each word type to a real vector in a continuous multidimensional space. Words which tend to occur in similar contexts in the training corpus appear in the vector space to either assemble in loosely structured relational groups, or to be located in particular geometric relationships or areas with respect to each other. For example, words involving a specific topic or words with noun-like relationships may be located in specific areas and be closely located to each other.

In addition to word-word contexts, we explore adding

- nouns,
- nouns and verbs (NV),
- Brown clusters [21], and
- Brown clusters with nouns and verbs (BNV)

as contexts to the co-occurrence matrix. Co-occurrence counts for these additional contexts were optimised on the development set. English noun and verb groups were taken from the LOB corpus [22].

4.2. Querying word embedding models

We use the word embedding vectors to perform queries using cosine similarity. The words used as query arguments are taken from the code-switched (CS) bigram examples in the training set. Consider the two known CS bigrams *i- album* and *ama- advertisers*. We use *album* and *advertisers* as query arguments to determine "most similar" candidate words for each. Table 6 lists the 10 most similar words for each query word. All but one of these are valid as possible CS bigrams. The CS bigram, *ama- ou*, is unlikely since *ou*, an Afrikaans slang term for *guy*, is a singular noun, but is preceded by a plural prefix.

Table 6: *Ten query results for words most similar to* album *and* advertisers *(sim.wrd: similar word; cos.scr: cosine similarity).*

Trigger	*i-*		*ama-*	
Query word	*album*		*advertisers*	
	sim.wrd	*cos.scr*	*sim.wrd*	*cos.scr*
Results	song	0.874	promoters	0.773
	movie	0.806	creatives	0.733
	film	0.774	employers	0.718
	soundtrack	0.771	sponsors	0.718
	series	0.736	fans	0.704
	footage	0.703	journalists	0.694
	gig	0.696	tenants	0.686
	animated	0.689	ou	0.682
	record	0.681	musicians	0.673
	book	0.672	spreadsheets	0.661

The number of query results generated per query word is optimised on the development set. All query results per trigger type are pooled. It may happen that a word appears multiple times in this pool. For example, the word *song* may result from query words *album*, *dance* and *celebration*, in each instance with a different similarity score. Methods of score combination and selection are investigated in the following section.

4.3. Synthesis of code-switched bigrams

New Zulu-to-English bigrams (ZEBGs) are synthesised by appending the result words to the trigger type of the particular query word, e.g. *i- song* (see Table 6). The similarity scores are used to synthesise pseudocounts for the new ZEBGs. If more than one instance of a word is present in the pool for a trigger type, the similarity scores are combined into a single pseudocount per ZEBG. We compare four combination functions.

- sum: If more than one cosine similarity is present for a result word, take the sum of the similarities as the pseudocount.
- const: Use a constant value of 1 as pseudocount.
- max: If more than one cosine similarity is present for a result word, take the maximum as the pseudocount.
- avg: If more than one cosine similarity is present for a result word, take the average as the pseudocount.

4.4. Language model training

We use the SRILM toolkit [23] for language modelling and evaluation. Bigram models were used throughout, with a vocabulary that was closed with respect to the development and test sets. Witten-Bell was found to outperform other discounting methods for our very sparse data. The baseline model is trained on the English-isiZulu CS corpus training set transcriptions. Augmented models are trained on the same data augmented with synthesised ZEBGs. A global scaling factor is applied to the pseudocounts of the synthesised ZEBGs and optimised on the development set. Interpolated versions of the baseline model and the optimum augmented model are trained by interpolating each main model with a secondary monolingual English bigram model (weight 0.1) and a secondary monolingual isiZulu bigram model (weight 0.1). Both secondary models are trained on web text.

4.5. Language model evaluation

Word embedding models are trained for a variety of parameter values. From each of these models, several sets of synthesised CS bigrams are generated. Each set is used to augment the language model (LM) training data, and a LM is trained. The LM resulting in the lowest development set perplexity is chosen as the optimum augmented model and is evaluated on the test set.

Besides reporting the usual language model perplexities (PP), we define a code-switch perplexity (CSPP). The CSPP is calculated only across Zulu-to-English code-switch boundaries, and therefore specifically indicates the confusability when switching languages.

5. Results and Discussion

We first consider the effect that the word embedding (WE) context window length, the number of WE dimensions and the pseudocount methods have on the perplexities (PPs). We average the PPs over a number of LMs to obtain reliable relative measures. Therefore the results in Tables 7 and 8 are for relative comparison for each parameter in question, and not for comparison against our baseline. Table 7a) shows three window lengths with corresponding average PPs calculated across 180 LMs. We see that the context window length has a weak effect on PP. Table 7b) shows results for three choices of dimension. Again, average PPs were calculated over 180 LMs for each dimensionality. Forty dimensions result in the lowest PPs, although this optimum may possibly change for a larger training set.

Table 7: *a) The average perplexities (PPs) across 180 LMs for word embedding context window lengths 3, 5 and 7. b) The average PPs across 180 LMs for word embedding dimensions 40, 100 and 200.*

a)	Window length	3	5	7
	PP dev	597.64	597.85	597.85
	PP test	819.40	819.40	819.70
b)	WE dimensions	40	100	200
	PP dev	596.76	598.33	598.25
	PP test	819.41	819.47	819.62

Table 8: *The average perplexities (PPs) across 135 LMs for pseudocount methods sum, const, max and avg.*

Method	sum	const	max	avg
PP dev	595.87	598.28	598.32	598.64
PP test	818.33	818.78	820.33	820.56

Table 8 compares the four proposed methods of determining pseudocounts. Each perplexity is an average over 135 LMs. The *sum* method results in the lowest perplexities.

We use the development set optimum values of 3 and 40 for the word embedding context window length and dimension, respectively, in an effort to reduce the hyperparameter search space in the experiments that follow. With these choices, we evaluate the context features used during WE model training.

Table 9 displays the perplexities of the baseline and the various augmented and interpolated language models, with best performance on the development set for the various ZEBG synthesis methods. For every synthesis method considered, the augmented LMs are able to improve on the baseline. Note how CSPPs are especially high, considering the 11k vocabulary size. This confirms the high confusability at a language switch. Disregarding the interpolated LMs, the best relative improvement, with respect to the baseline, is afforded by the LM using parts-of-speech classes and Brown clusters (BNV+const), and leads to a 7.29% decrease in CSPP for the test set. Interpolation accounts for a 6.9% relative decrease in test set PP when compared with the baseline. A further 5.86% relative improvement in CSPP is achieved when augmenting the interpolated baseline with synthetic ZEBGs, yielding the best result (BNV+const+i).

Most '+sum' LMs slightly outperform their '+const' counterparts on the development set, while the '+const' LMs consistently outperform '+sum' LMs on the test set. This indicates that '+const' models tend to generalise better than '+sum' models, which appear to specialise on the development set.

Figure 2 shows how PP changes as the number of queries per query word is varied. We observe the general trend that the PP increases as the number of queries increase. The two 'const' methods give PPs lower than the baseline for query sizes less than 15.

6. Conclusions and future work

We have shown that word embeddings can be used to successfully synthesise Zulu-to-English code-switch bigrams. Despite the small size of our corpus, which was compiled from South African soap opera speech, we were able to find for each synthesis method an optimum model giving reduced perplexity by including these synthesised bigrams during language model training. By considering the perplexity specifically across the language switch, we could show that much greater confusability exists at the location of the language change, and that this confusability is reduced by the augmentation with synthesised bigrams. Analysis of our data revealed that code-switches may occur mid-word, when isiZulu prefixes are joined to English words. The analysis of the code-switch examples also indicated some semantic and syntactic regularity, and we show that this can be taken advantage of by including parts-of-speech or automatically-determined word classes in the embedding process. Besides the inclusion of parts-of-speech, the synthesis procedure is language independent and is being extended to other South African languages with code-switching such as English-isiXhosa and English-Setswana in ongoing work.

Table 9: *The full perplexities (PPs) and Zulu-to-English code-switch perplexities (CSPP) for the development and test sets for the various WE context features and pseudocount combination methods. Perplexities are given for a) baseline, b) augmented, c) interpolated baseline, and d) interpolated augmented LMs. The best results for the augmented (BNV+const) and interpolated augmented LMs are in bold. (NV: Noun and verb contexts included in WE; BNV: Brown clusters, noun and verb contexts included in WE; '+i': Interpolated LM)*

	LM	PP		CSPP	
		Dev	**Test**	**Dev**	**Test**
a)	Baseline	595.33	818.78	2696.39	3163.08
b)	sum	591.03	815.12	2510.76	3057.42
	const	591.66	814.01	2498.25	2971.04
	Nouns+sum	590.49	816.26	2500.16	3098.38
	Nouns+const	591.66	814.01	2498.25	2971.04
	NV+sum	591.15	817.74	2463.89	3088.74
	NV+const	592.68	814.16	2538.19	2975.85
	Brown+sum	590.88	816.63	2502.83	3102.89
	Brown+const	590.36	814.21	2465.67	2991.88
	BNV+sum	590.37	815.78	2494.30	3080.61
	BNV+const	591.36	**812.77**	2486.50	**2932.63**
c)	Baseline+i	525.57	762.19	2619.62	2953.26
d)	BNV+sum+i	522.01	760.40	2459.35	2907.72
	BNV+const+i	522.68	**757.68**	2453.19	**2780.24**

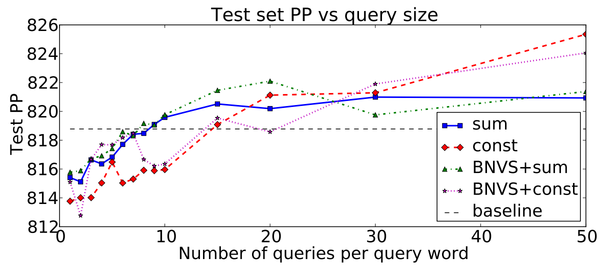

Figure 2: *The test set PP versus the number of query results per query word. The horizontal dashed line is the baseline test PP.*

7. Acknowledgements

We would like to thank e.tv and Yula Quinn at Rhythm City for assistance with data compilation.

8. References

[1] J. Franco and T. Solorio, "Baby-steps towards building a Spanglish language model," *Computational Linguistics and Intelligent Text Processing*, p. 75, 2007.

[2] T. Solorio and Y. Liu, "Learning to predict code-switching points," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, p. 973.

[3] J. Y. C. Chan, H. Cao, P. C. Ching, and T. Lee, "Automatic recognition of Cantonese-English code-mixing speech," *International Journal of Computational Linguistics and Chinese Language Processing*, vol. 14, no. 3, p. 281, 2009.

[4] H. Cao, P. C. Ching, T. Lee, and Y. T. Yeung, "Semantics-based language modeling for Cantonese-English code-mixing speech recognition," in *Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on*, 2010, pp. 246–250.

[5] K. K. Bhuvanagiri and S. K. Kopparapu, "An approach to mixed language automatic speech recognition," in *Oriental COCOSDA 2010*, 2010. [Online]. Available: http://desceco.org/O-COCOSDA2010/proceedings.php

[6] T. I. Modipa, M. H. Davel, and F. de Wet, "Implications of Sepedi/English code switching for ASR systems," in *PRASA 2013 Proceedings*, 2013.

[7] C.-F. Yeh, Y.-C. Lin, and L.-S. Lee, "Minimum phone error model training on merged acoustic units for transcribing bilingual code-switched speech," in *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*, 2012, pp. 320–324.

[8] W.-B. Liang, C.-H. Wu, and C.-S. Hsu, "Code-switching event detection based on Delta-BIC using phonetic eigenvoice models," in *14th Annual Conference of the International Speech Communication Association*, 2013, p. 1487.

[9] C.-J. Shia, Y.-H. Chiu, J.-H. Hsieh, and C.-H. Wu, "Language boundary detection and identification of mixed-language speech based on map estimation," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 1, 2004, pp. I–381–4 vol.1.

[10] H. Adel, N. T. Vu, F. Kraus, T. Schlippe, H. Li, and T. Schultz, "Recurrent neural network language modeling for code- switching conversational speech," in *ICASSP 2013*, vol. ICASSP 2013, 2013.

[11] H. Adel, N. T. Vu, and T. Schultz, "Combinations of recurrent neural networks and factored language models for code-switching language modeling," in *ACL 2013*, vol. ACL 2013, 2013.

[12] H. Adel, N. T. Vu, K. Kirchhoff, D. Telaar, and T. Schultz, "Syntactic and semantic features for code-switching factored language models," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 3, pp. 431–440, 2015.

[13] Y. Li, P. Fung, P. Xu, and Y. Liu, "Asymmetric acoustic modeling of mixed language speech," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 5004–5007.

[14] Y. Li and P. Fung, "Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 7368–7372.

[15] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, p. 211, 2015.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2013.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[18] E. van der Westhuizen and T. R. Niesler, "Automatic speech recognition of English-isiZulu code-switched speech from South African soap operas," in *Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2016.

[19] G. Poulos and C. T. Msimang, *A Linguistic Analysis of Zulu*, 1st ed. Pretoria: Via Afrika, 1998.

[20] M. Kurimo, A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraçlar, "Unlimited vocabulary speech recognition for agglutinative languages," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2006.

[21] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[22] S. Johansson, G. N. Leech, and H. Goodluck, "Manual of information to accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital computers." [Online]. Available: http://clu.uni.no/icame/manuals/LOB/INDEX.HTM

[23] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. Intl. Conf. Spoken Language Processing*, 2002.