Maximum Entropy Language Modeling for Russian ASR

Evgeniy Shin, Sebastian Stüker, Kevin Kilgour,
Christian Fügen, Alex Waibel

International Center for Advanced Communication Technology
Institute for Anthropomatics, Karlsruhe Institute of Technology
Karlsruhe, Germany

Abstract

Russian is a challenging language for automatic speech recognition systems due to its rich morphology. This rich morphology stems from Russian's highly inflectional nature and the frequent use of pre- and suffixes. Also, Russian has a very free word order, changes in which are used to reflect connotations of the sentences. Dealing with these phenomena is rather difficult for traditional n-gram models. We therefore investigate in this paper the use of a maximum entropy language model for Russian whose features are specifically designed to deal with the inflections in Russian, as well as the loose word order. We combine this with a sub-word based language model in order to alleviate the problem of large vocabulary sizes necessary for dealing with highly inflecting languages. Applying the maximum entropy language model during re-scoring improves the word error rate of our recognition system by 1.2% absolute, while the use of the sub-word based language model reduces the vocabulary size from 120k to 40k and the OOV rate from 4.8% to 2.1%.

1. Introduction

The Russian language has some properties that make the creation of high performing *Large Vocabulary Continuous Speech Recognition* (LVCSR) quite challenging. Especially in language modeling there are two principal problems that need to be dealt with:

- *Morphology:* Russian is a highly inflecting language. E.g., Russian nouns can be declined according to six cases, two numbers (singular and plural) and three grammatical genders (male, female and neutral). Adjectives need to declined in accordance with the subject that they belong to; verbs can be conjugated according to three persons, two numbers and two tenses. Prefixes and suffixes are frequently used to produce a multitude of derivatives of basic words.

- *Word Order:* The word order in Russian is rather free. Different word orders for the same sentence are used to convey different connotations.

The rich morphology of Russian leads to the need for large vocabularies. And even with rather large vocabularies ASR systems suffer from relatively high *out of vocabulary* (OOV) rates [1, 2].

Also, the combination of loose word order and rich morphology leads to very high perplexities for standard n-gram language models, especially when trained estimated on moderate amounts of training data [1, 3]. Larger vocabularies generally lead to higher n-gram language model perplexities. The same is true for the loose word order, as n-gram language models compose the sentence language model probability from the probabilities of word sequences of fixed order and short length.

In order to deal with the problem of high OOV rates that arise from the rich morphology of a language, the use of sub-word based search vocabularies is a common technique and has been successfully used in a multitude of languages (see Section 2). However, their impact on the problems of the high perplexities of the language model are only limited, especially for Russian with respect to its many endings arising from the grammatical inflections, but also with respect to its many prefixes and suffixes that can be combined with a myriad of words.

In order to alleviate this problem we propose the application of maximum entropy language models to Russian. In this paper we present an implementation of such a maximum entropy language model that deals specifically with the phenomena that make n-gram language models perform badly for Russian. We combine the maximum entropy model with our implementation of a sub-word based vocabulary and evaluate both approaches on a large vocabulary continuous speech recognition task in the tourist domain.

The rest of the paper is structured as follows. In Section 2 we give an overview of related work in both areas – sub-word based language modeling and maximum entropy language models. Section 3 then introduces our approach to sub-word based language modeling for Russian, while Section 4 describes our design of an entropy based language model that deals specifically with Russian morphology. In Section 5 we report on the improvements in word error rate that we achieved with the approaches described in this paper.

2. Related Work

2.1. Sub-Word Based Language Models

Sub-word based language models have been reported to be successful for highly inflecting languages such as Russian[4, 1], Czech[5], Finnish[6], Turkish[7], Slovenian[8], Arabic[9, 10].

In [9] *SyntaxNN*, a neural network language model using syntactic and morphological features, and *DLM*, a discriminative language model trained using the Minimum Bayes Risk (MBR) criterion, and unigram, bigram, and trigram morphs features were applied to Arabic.

To incorporate syntactical and morphological knowledge of Arabic to language modeling [10] utilized a *Factored Language Modeling* toolkit[11]. The use of word lexeme and morpheme features led to a reduction in WER of 2% relative.

A particle (similar to sub-word) based n-gram model in combination with a word based model applied to Russian was shown to give a reduction of perplexity of up to 7.5% [4]. For this, data-driven techniques were applied that determine particle units and word decompositions automatically.

A random-forest language model for Russian[4] using word stems among other morphological features achieved a WER improvement of 3.4% relative over a trigram model.

[12] explored the use of sub-word based language models for Finnish, Estonian, Turkish and Egyptian Colloquial Arabic. They performed word decomposition in an unsupervised, data-driven way using *Morfessor*. They showed that the morph models performed fairly well on OOVs without compromising the recognition accuracy of in-vocabulary words.

An application of sub-word based language model to Czech is studied in [5]. A sub-word based language model which includes different models for different sub-word units, such as stems and endings, reduces the WER by about 7% absolute. They applied their language model in n-best list re-scoring.

An interesting idea is proposed in [7]. Here, Turkish was modeled with so called FlexGrams, which allow skipping several parents and use later grams in the history to estimate a probability of the current word. They experimented with words split into their stem and suffix forms, and defined stem-suffix FlexGrams, where one set of offsets is applied to stems and another to suffixes.

2.2. Maximum Entropy Language Models

The maximum entropy approach was introduced to language modeling more than 10 years ago[13, 14, 15]. And it is being used today the state-of-the-art language models such as ModelM[16].

ModelM[16] is an exponential class-based n-gram language model. The word n-gram and word class features are incorporated into the language model within an exponential modeling framework. The model with enhanced word classing[17] achives a total gain of up to 3.0% absolute over a Katz-smoothed trigram model[17]. Experiments were done on the Wall Street Journal corpus.

Maximum Entropy models are also being successfully used for machine translation systems, e.g. [18, 19]

In [19] it was shown that the use of discriminative word lexica (DWL) can improve the translation quality significantly. For every target word, they trained a maximum entropy model to determine whether this target word should be in the translated sentence or not. As features for their classifier they used one feature per source word.

3. Sub-Word Based Search Vocabulary and Language Model

The goal of sub-word based search vocabularies and language models is to reduce the OOV rate of an ASR system by decomposing whole words into smaller units. Normally, the distinct number of these sub-word units is significantly smaller than the number of words that they form. So, with constant vocabulary size, the OOV rate of the recognition system is drastically reduced.

In order to work, the following steps need to be taken:

- *Decomposition:* The original words need to be decomposed into smaller units. The units need to show some sort of consistency, so that their total number is clearly smaller than that of the words that they were derived from. Depending on the language one can decide to either decompose all words in the search vocabulary, or only a certain sub-set, e.g., those occurring relatively infrequently, while the frequent words are being kept intact. Word decomposition is usually done for the language model training material and then a new vocabulary is derived.

- *Pronunciation Generation:* For the generated sub-word units pronunciations need to be added to the system's dictionary. Since in general the mapping between the writing of a word and its pronunciation, i.e. phoneme sequence, is not given or easily derivable, deducting the pronunciation of the sub-word units from the pronunciation of the original words is often not straight-forward or even impossible. Often grapheme based pronunciation dictionaries can offer a solution here.

- *Language Model Training:* Based on the new vocabulary composed of the sub-word units, and potentially mixed with whole words, a new language model needs to be trained that is then used for recognition.

- *Word Reconstruction:* After decoding, the recognized sub-words need to be recombined in order to obtain a valid word sequence.

3.1. Word Decomposition and Merging

For word decomposition we used a *Snowball* [20] based stemmer. Snowball is a small string processing language designed for creating stemming algorithms. A stemmer for Russian is distributed with the package. The stemmer is not a tool for morpheme analysis, but a word stem derivation tool. Therefore, the output of this tool needs to be processed to split up words into subunits. For a given word the stemmer returns a stem. Endings can then be derived by comparing the original word string against that of the stem. For example the words in the phrase "необходимое условие" (necessary conditions) are decomposed into:

word		stem		ending
необходимое	→	необходим	→	ое
условие	→	услов	→	ие

Compound words that are joined via a hyphen, are first split before being put through the stemmer, as every sub part of a compound might have its own ending.

In order to simplify the merging of sub-words after decoding every word part after the first stem is marked as an ending. After decoding all endings after a stem are merged to the stem, until a new stem is encountered. For words that do not have an explicit ending, the null-ending was utilized for language modeling.

4. Maximum Entropy Language Modeling

In maximum entropy modeling the model is constrained by features. In language modeling these features must be extractable from the word sequence for which the probability needs to be calculated. The models are then trained according to the maximum conditional entropy criterion. Thereby a number of different training algorithms are available for finding the probability distribution with the maximum entropy, given the training data.

4.1. Features

For n-gram models the features used are the bigrams, trigrams, etc. that appear in the word sequence. For maximum entropy language models one can use additional features, such as part of speech (POS) tags, different grammatical categories or topic information. All these kinds of features can be represented by binary feature functions or indicator functions.

A bigram feature can for example be expressed by the following indicator function:

$$f_1(x,y) = \begin{cases} 1, & if \quad y = "day" \quad and \quad x = "nice" \\ 0, & otherwise \end{cases}$$

The function, feature respectively, f_1 returns 1 for the word y and its context x, if y and x form the bigram "nice day".

Using large amounts of training data we can estimate the probability distribution $p_e(x,y)$ where x and y can take on all possible words in the search vocabulary. Now, with the help of p_e, we can estimate a mean value of feature f_1:

$$\mu(f_1) = \sum_{x,y} p_e(x,y) f_1(x,y) = \sum_{x,y} relfreq(x,y) f_1(x,y) \tag{1}$$

If the training data is sufficiently large, the mean value represents the expected value of the real distribution:

$$\mathbb{E}(f_1) = \sum_{x,y} p(x,y) f_1(x,y) \tag{2}$$

Our language model p_m is requested to be unbiased with respect to f_1, i.e. to have the same expected value for the feature f_1:

$$\sum_{x,y} p_e(x,y) f_1(x,y) = \sum_{x,y} p_m(x,y) f_1(x,y), \tag{3}$$

where $p_m(x,y)$ is the distribution as given by the model.

However, we are interested in modeling $p(y|x)$ and not $p(x,y)$. Therefore the constraint equations for feature f_1 has to be:

$$\sum_{x,y} p_e(x,y) f_1(x,y) = \sum_{x,y} p_e(x) p_m(y|x) f_1(x,y), \tag{4}$$

For every feature that we define for the maximum likelihood model such a constraint function is defined and has to be obeyed by our model distribution p_m.

4.2. Maximization of conditional entropy

Depending on which features we select for our language model, not only one but a whole set of distributions that comply with the constraints exists. From these many possible distributions the best one needs to be selected. One approach comes from information theory and is based on the concept of conditional entropy:

$$H(Y|X) = - \sum_{\substack{x \in X, \\ y \in Y}} p(x,y) \log p(y|x) \tag{5}$$

The idea of maximum entropy modeling is to choose that model which maximizes the conditional entropy of labels y given an information x (e.g., word context):

$$p_{me} = \arg\max_{p_m} H(p_m) \tag{6}$$

In simple words this means that the model makes no further assumptions about the given features. With the help of Lagrange multipliers, which are used to solve this constrained optimization problem, it can be shown that the resulting probability distribution has the parametric form:

$$p_{me}(\lambda) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(x,y) \right), \tag{7}$$

where $f_i(x,y)$ are binary feature functions. λ_i are weight factors—parameters of the model. $Z(x)$ is the normalization factor in order to ensure that result is indeed a probability distribution.

4.3. Training

A number of algorithms can be used for estimating the parameters of a maximum entropy model. There are both—–special methods, such as *Generalized Iterative Scaling*[21], *Improved Iterative Scaling*[22], and general purpose optimization techniques, such as gradient ascent, conjugate gradient and quasi-Newton methods. [23] in its comparison of algorithms for maximum entropy parameter estimation states that the widely used iterative scaling algorithms perform quite poorly, and for all of the test problems, a limited memory variable metric algorithm outperformed the other choices.

Four our experiments we used *Limited-memory BFGS* a limited memory variation of the *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) method [24, 25], which is an implementation of the variable metric method. For this we used the *CRF++* Toolkit[26].

5. Experimental Set-Up and Results

We evaluated our two approaches on Russian data that was recorded by Mobile Technologies in the domain of tourist and basic medical needs, as it can be found in mobile speech translation devices such as Jibbigo[1]. We compare our results to a baseline with a word based n-gram model, while we keep the acoustic model fixed.

5.1. Data Set

The acoustic model training data accounts for about 620 hours of broadcast news and broadcast conversations acquired within the *QUAERO*[27] project. Further, we used a data set of read speech mostly in touristic and medical speech domains, provided by *Mobile Technology GmbH*[28]. From this set of 63 hours we cut away 3 hours as test set, while the rest went into acoustic model training.

For training our language models we used a text corpus collected from the Internet, 156M tokens in size. The text was crawled from forums in the touristic and medical domain.

The word decomposition for the sub-word based as well as the maximum entropy language model was done with the Snowball stemming algorithm[20].

Table 1 gives an overview for the datasets used.

AM training	Broadcast news & radio	620 hours
AM training	Read speech	60 hours
LM training	Web forums	156M words
Testing	Read speech	3 hours

Table 1: Over view over the acoustic data used for testing and AM training

5.2. Baseline System

We performed all experiments with the help of the Janus Recognition Toolkit featuring the IBIS single pass decoder [29]. For our HMM based acoustic model we used a context dependent quinphone setup with three states per phoneme, and a left-to-right topology without skip states. The 8,000 models of the HMM were trained using *incremental splitting of Gaussians* (MAS) training, followed by *optimal feature space* training and 2 iterations of Viterbi training. The models were further improved with boosted MMIE training [30].

For the baseline system we used a standard 4-gram language model which we trained with the help of the SRI LM toolkit [31]. The search vocabulary was taken from the 120k most frequent words from the LM training data. For both cases the dictionaries are grapheme based dictionaries which works quite well for Russian [3].

5.3. Sub-Word Based Experiments

The sub-word based system uses a sub-word search vocabulary and a sub-word based 4-gram model. For this we split the words in the language model training with our procedure described in Section 3. As vocabulary we selected the 40k most frequent sub-word units.

5.4. Re-Scoring with Word N-Gram Model

While sub-word based language modeling reduces the OOV rate, it introduces additional problems such as a loss in language model reach, and the fact that the sub-word units are acoustically more confusable. Therefore, in order to combine the advantages of a sub-word based and a word based LM we re-scored n-best lists that were generated with the sub-word based LM.

Re-scoring was done by interpolating the combined acoustic and LM model scores of the sub-word based system with the LM score from the word based 4-gram LM. Interpolation was done as a weighted sum of the scores in the log domain. We tested a series of interpolation weights from 0 to 10.

5.5. Re-Scoring with Maximum Entropy LM

Word endings in Russian depend on several grammatical features of the current word, such as gender, case, tens, and form a pattern for the utterance. At the same time recognizing the endings correctly is quite challenging, as they have little acoustic evidence and are difficult to model with a regular n-gram LM. So, we selected features for the maximum entropy model that help with discriminating the endings. The features consist of words and endings in their context. Here

[1] http://www.jibbigo.com

is a small example:

s_{-5}	e_{-5}	как	~#
s_{-4}	e_{-4}	подчеркнул	~#
s_{-3}	e_{-3}	офицер	~#
s_{-2}	e_{-2}	полиц	~ии
s_{-1}	e_{-1}	жёстк	~ие
s_0	e_0	мер	~ы
s_1	e_1	не	~#
s_2	e_2	применя	~лись

Since applying the entropy language model during regular decoding is too computationally intensive, again we applied the language model during n-best list re-scoring. For calculating the LM score we used the three previous stems (s_{-3}, s_{-2}, s_{-1}), three previous endings (e_{-3}, e_{-2}, e_{-1}) and one successor stem (s_1) and ending (s_1) as features. The null ending is explicitly modeled with the ~# place-holder.

For training, the *CRF++* Toolkit[26] is utilized. As the training of the labels, endings in our case, within a single model was not possible due to main memory usage (more than 512GB RAM was needed), a similar approach as in [18] and [19] was applied. The idea is to train a separate model for every label. Every model evaluates then only two classes: the ending, which the models stands for versus all other endings.

In testing, all models, whose corresponding endings were present in the utterance, were applied. The resulting score is given by the sum of the scores from the single models.

Again we re-scored the n-best lists generated by the sub-word system by interpolating the language model score from the maximum entropy language model with the combined acoustic and LM scores from the sub-word system. As for the interpolation described above we tested a series of interpolation weights, this time in the range of 0 to 20.

5.6. Results

5.6.1. Baseline System and Sub-Word Based System

Table 2 shows results of the full-word baseline and the sub-word based system. It can be seen that in spite of the fact that the OOV rate of the full-word system (4.8%) is higher than that of the sub-word system (2.1%), the latter performs slightly worse. Two of the reasons for that could be the higher acoustic confusability between the shorter sub-words and the shorter context of the sub-word based n-gram language model. The OOV rate of the sub-word based system is quite high but still half of that of the full-word system. The reason for that could be the difference in vocabulary size (40k vs. 120k).

	WER	OOV	vocabulary size
baseline	25.7%	4.8%	120k
sub-words	25.9%	2.1%	40k

Table 2: Word error rates, OOV rates and vocabulary sizes of the word based baseline and the sub-word based system

5.6.2. Re-Scoring with Word Based LM and Maximum Entropy LM

Figure 1 shows the result of our experiments in re-scoring the n-best lists from the sub-word system with a series of interpolation weights. One can see that for re-scoring with the word based LM, when choosing the right interpolation weight, we can improve the WER of the sub-word based system by 0.4% absolute.

When re-scoring with the maximum entropy model we can improve the WER of the sub-word based model by up to 1.2% absolute. We can also see that the interpolation is rather insensitive to the interpolation weight Finally, we combined both language models in the interpolation during re-scoring, taking the best interpolation weights from the individual re-scoring experiments. Table 3 shows the results of this combination. We can see that the improvements from the two language models sum up, i.e. their gains seem to be orthogonal to each other. In that way we can reduce the WER of the sub-word based based system by 1.6% absolute and that of our baseline system with the word based n-gram LM by 1.4% absolute.

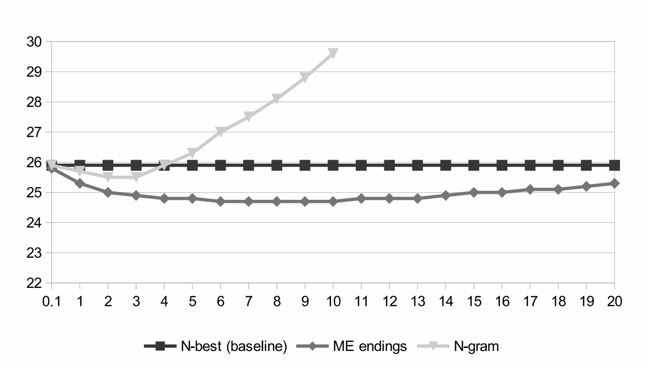

Figure 1: WER of re-scoring the n-best list of the sub-word system with the full word 4-gram model and with the maximum entropy model using different interpolation weights

Baseline	25.7%
Subwords	25.9%
+ Maximum entropy	24.7%
+ Word n-gram	24.3%

Table 3: Combined results of recognition and re-scoring systems

6. Conclusion

In this paper we investigated the use of a maximum entropy language model in order to deal with the highly inflectional nature of Russian and its loose word order. We designed the features of the language model specifically to target these problems. Applying the maximum entropy model during n-best list rescoring reduces the word error rate of our baseline system by 1.2% absolute. In order to deal with the need for a large vocabulary for a Russian ASR system due to the many inflections possible in Russian, we implemented a sub-word based LM based on stemming. Using this language model reduces the vocabulary necessary during decoding from 120k to 40k and the OOV rate from 4.8% to 2.1%. By re-scoring the n-best lists of the sub-word based system with a combination of the maximum entropy language model and a word based 4-gram model, we can reduce the word error rate by another 0.2% absolute.

7. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No 287658 — 'Bridges Across the Language Divide' (EU-BRIDGE). Also, this work was in part realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. 'Research Group 3-01' received financial support by the 'Concept for the Future' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

8. References

[1] E. Whittaker, "Statistical language modelling for automatic speech recognition of russian and english," *Daktaro disertacija, Cambridge University Engineering Department, Cambridge*, 2000.

[2] Y. Titov, K. Kilgour, S. Stüker, and A. Waibel, "The 2011 kit quaero speech-to-text system for the russian language," in *Proceedings of the 14th International Conference "Speech and Computer" (SPECOM'2011)*, September 2011.

[3] S. Stüker and T. Schultz, "A grapheme based speech recognition system for russian," in *Proceedings of the 9th International Conference "Speech And Computer" SPECOM'2004*. Saint-Petersburg, Russia: Anatolya, September 2004, pp. 297–303.

[4] I. Oparin, "Language models for automatic speech recognition of inflectional languages," Ph.D. dissertation, University of West Bohemia, 2008.

[5] P. Ircing, P. Krbec, J. Hajic, J. Psutka, S. Khudanpur, F. Jelinek, and W. Byrne, "On large vocabulary continuous speech recognition of highly inflectional language-czech," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[6] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to finnish," *Computer Speech & Language*, vol. 20, no. 4, pp. 515–541, 2006.

[7] D. Yuret and E. Biçici, "Modeling morphologically rich languages using split words and unstructured dependencies," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, 2009, pp. 345–348.

[8] T. Rotovnik, M. Maucec, and Z. Kacic, "Large vocabulary continuous speech recognition of an inflected language using stems and endings," *Speech communication*, vol. 49, no. 6, pp. 437–452, 2007.

[9] L. Mangu, H. Kuo, S. Chu, B. Kingsbury, G. Saon, H. Soltau, and F. Biadsy, "The ibm 2011 gale arabic speech transcription system," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 272–277.

[10] A. El-Desoky, R. Schlüter, and H. Ney, "A hybrid morphologically decomposed factored language models for arabic lvcsr," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 701–704.

[11] K. Kirchhoff, J. Bilmes, and K. Duh, "Factored language models tutorial," 2007.

[12] M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A. Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 5, no. 1, p. 3, 2007.

[13] A. Berger, V. Pietra, and S. Pietra, "A maximum entropy approach to natural language processing," *Computational linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[14] R. Rosenfield, "A maximum entropy approach to adaptive statistical language modeling," 1996.

[15] R. Rosenfeld, S. Chen, and X. Zhu, "Whole-sentence exponential language models: a vehicle for linguistic-statistical integration," *Computer Speech & Language*, vol. 15, no. 1, pp. 55–73, 2001.

[16] S. Chen, "Shrinking exponential language models," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 468–476.

[17] S. Chen and S. Chu, "Enhanced word classing for model m," in *Proceedings of Interspeech*, 2010, pp. 1037–1040.

[18] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The kit english-french translation systems for iwslt 2011," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[19] A. Mauser, S. Hasan, and H. Ney, "Extending statistical machine translation with discriminative and trigger-based lexicon models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 210–218.

[20] M. Porter, "Snowball: A language for stemming algorithms," 2001.

[21] J. N. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *The annals of mathematical statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.

[22] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 4, pp. 380–393, 1997.

[23] R. Malouf *et al.*, "A comparison of algorithms for maximum entropy parameter estimation," in *Proceedings of the sixth conference on natural language learning (CoNLL-2002)*, 2002, pp. 49–55.

[24] M. Avriel, *Nonlinear programming: analysis and methods*. Courier Dover Publications, 2003.

[25] J. F. Bonnans, *Numerical optimization: theoretical and practical aspects: with 26 figures*. Springer-Verlag New York Incorporated, 2003.

[26] T. Kudo. (2005, Apr.) Crf++: Yet another crf tool kit. [Online]. Available: http://code.google.com/p/crfpp/

[27] (2008, Mar.) Quaero is a european research and development program. [Online]. Available: http://www.quaero.org/

[28] Mobile technologies gmbh. [Online]. Available: http://www.jibbigo.com/

[29] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A One Pass-Decoder Based on Polymorphic Linguistic Context Assignment," in *ASRU*, Madonna di Campiglio Trento, Italy, December 2001.

[30] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4057–4060.

[31] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit," in *Proceedings of the international conference on spoken language processing*, vol. 2, 2002, pp. 901–904.