Stem-based Maximum Entropy Language Models for Inflectional Languages

Dimitrios Oikonomidis and Vassilios Digalakis

Technical University of Crete, Greece
{doikon,vas}@telecom.tuc.gr

Abstract

In this work we build language models using three different training methods: n-gram, class-based and maximum entropy models. The main issue is the use of stem information to cope with the very large number of distinct words of an inflectional language, like Greek. We compare the three models with both perplexity and word error rate. We also examine thoroughly the perplexity differences of the three models on specific subsets of words.

1. Introduction

Greek is an inflectional and morphologically rich language. Inflectional rules primarily mark verbs for tense and agreement with subjects and objects in number, gender and person and mark nouns and adjectives for case and number.

Verb tenses are formed by inflection: a *prefix* and/or *suffix* modify a *root* and change the tense. For example the tenses of verb *τρέχω* (I run) are

Simple Present	τρέ-χω	I run
Present Perfect	έχω τρέ-ξει	I have run
Simple Past	έ-τρε-ξα	I ran
Past Cont.	έ-τρε-χα	I was running
Past Perfect	είχα τρέ-ξει	I had run
Simple Future	θα τρέ-ξω	I will run
Future Cont.	θα τρέ-χω	I will be running
Future Perfect	θα έχω τρέ-ξει	I will have run

Table 1: Tenses of verb *τρέχω* (I run)

Moreover, each tense has six persons, which are formed by modifying the suffix. For example, the simple present of verb *τρέχω* (I run) is

τρέχ-ω	I run
τρέχ-εις	you run
τρέχ-ει	he, she, it runs
τρέχ-ουμε	we run
τρέχ-ετε	you run
τρέχ-ουν	they run

Table 2: Simple present of verb *τρέχω* (I run)

There are eight tenses in Greek and each one has six persons. Thus, the total number of forms a verb can have is forty eight. In English a regular verb has only four distinct forms and irregular verbs have at most eight forms. These are

- the root or base form: e.g. walk
- the third singular present tense: e.g. walks
- the gerund and present participle: e.g. walking
- the past tense and past/passive participle: e.g. walked

Articles, adjectives and nouns must agree with the subject in number, gender and person. For example, the phrase *ο μαύρος σκύλος* (the black dog) has the following cases for singular and plural tenses

Case	Singular	Plural
nominative	ο μάυρ-ος σκύλ-ος	οι μαύρ-οι σκύλ-οι
genitive	του μαύρ-ου σκύλ-ου	των μαύρ-ων σκύλ-ων
accusative	τον μαύρ-ο σκύλ-ο	τους μαύρ-ους σκύλ-ους
dative	μαύρ-ε σκύλ-ε	μαύρ-οι σκύλ-οι

Table 3: Cases of a singular adjective/noun

English has only one inflection of the noun, the plural form, which is usually formed by appending the suffix –s, e.g. dog: dog-s.

Moreover, there are genders. The dog is masculine, the cat is feminine and the horse is neuter. Adjectives must agree with nouns. Compare the following table which shows the cases for the phrase *η μαύρη γάτα* (the black cat), where now cat is feminine, with the previous table where the dog was masculine

Case	Singular	Plural
nominative	η μάυρ-η γάτ-α	οι μαύρ-ες γάτ-ες
genitive	της μαύρ-ης γάτ-ας	των μαύρ-ων γατ-ών
accusative	την μαύρ-η γάτ-α	τις μαύρ-ες γάτ-ες
dative	μαύρ-η γάτ-α	μαύρ-ες γάτ-ες

Table 4: Cases of a feminine adjective/noun

All these forms produce a very big vocabulary for the Greek language. Table 5 summarizes some characteristics of the Greek and other three languages: English, French and German. The fourth row shows that Greek is somewhere between French and German in respect of the number of different words that exist in a corpus of about thirty five million words. German is by far the richest language and English the most impoverished. The sixth row shows the lexical coverage of each language. As can be seen, with sixty thousands words we cover 99.6% of English while with the same vocabulary we cover only 96.5% of Greek. A Greek vocabulary of 220 K words is needed in order to achieve 99.6% lexical coverage. English, French and German data are taken from [1].

	English	French	Greek	German
Source	WSJ	Le Monde	Elefthe rotypia	Frank furter
Corpus size	37.2 M	37.7 M	35 M	31.5M
Distinct words	165 K	280 K	410 K	500 K
Vocabulary size	60 K	60 K	60 K	60 K
Lexical coverage	99.6%	98.3%	96.5%	95.1%

Table 5: Characteristics of four languages

2. Back-off Language Models

For all the experiments in this work we used a corpus taken from the Eleftherotypia daily newspaper. We split the corpus into training and test sets containing 35 M and 3.5 M words, respectively. To investigate the effect of using different training sets, we split the training set into three sets: one containing 1 M words, one containing 5 M words and one with all the 35 M words. We built back-off [2] language models using four smoothing methods: Good-Turing [3], Witten-Bell [4], Absolute Discounting [5] and Modified Kneser-Ney [6] and evaluated perplexity on the same test set of 3.5 M words. All models had a trigram cutoff threshold of 2 (e.g singleton trigrams were excluded). The vocabulary was 60K words. Additionally, we run speech recognition experiments using a Gaussian-mixture, hidden Markov model (HMM) speech recognizer. Details about the acoustic model can be found in [7] The recognizer was tuned so that it recognized in 5x real time on a Pentium II 400MHz processor. The test set for speech recognition experiments was comprised of 8 different speakers and 800 utterances. The total number of words was 13863. Table 6 summarizes the results.

	1M text		5M text		35M text	
Smoothing	PP	WER	PP	WER	PP	WER
Good-Turing	341	27.71	248	23.48	163	19.59
Witten-Bell	354	27.42	251	24.17	163	19.84
Absolute Discounting	344	28.47	256	24.25	169	20.78
Modified Kneser-Ney	328	26.78	237	21.91	156	18.57

Table 6: Perplexity and Word Error Rate of four smoothing methods

Table 7 shows the out-of-vocabulary word rate of the speech recognition test text for the three training sets.

	1M	5M	35M
OOV	4.75%	3.46%	3.17%

Table 7: OOV rate

Table 8 shows the hit rate of 1-, 2-, 3-grams, e.g. how many times the language model used a trigram or backed-off to bigram or unigram probability.

	hit rate (%) 1M	hit rate (%) 5M	hit rate (%) 35M
1-gram	27.3	16.4	7.4
2-gram	52.5	49.9	40
3-gram	20.2	33.7	52.6

Table 8: Hit rate of 1-, 2-, 3-grams

3. Class-based Models

We clustered the vocabulary into about 30000 classes based on the stem of words. Table 9 shows some of these classes.

- άγνωστ (unkown): άγνωστος, άγνωστου, άγνωστο, άγνωστοι, άγνωστους, άγνωστη, άγνωστης, άγνωστες, άγνωστα
- βλέπ (see): βλέπω, βλέπεις, βλέπει, βλέπουμε, βλέπετε, βλέπουν, βλέπαμε, βλέπατε, βλέποντας
- δήμαρχ (mayor): δήμαρχος, δήμαρχο, δήμαρχοι
- καθηγητ (professor): καθηγητής, καθηγητή, καθηγητές, καθηγητών
- καλύπτ (cover): καλύπτει, καλύπτουμε, καλύπτουν, καλύπτεται, καλύπτονται, καλύπτοντας
- πιστεύ (believe): πιστεύω, πιστεύεις, πιστεύει, πιστεύουμε, πιστεύετε, πιστεύουν

Table 9: Classes based on stem

We built class-based language models of the form

$$p(w_i \mid w_{i-2}, w_{i-1}) = p(g_i \mid g_{i-2}, g_{i-1})\, p(w_i \mid g_i) \quad (1)$$

Table 10 shows the perplexity of the class-based model for the three training sets

Model	PP (1M)	PP (5M)	PP (35M)
Class-based	383	299	215

Table 10: Perplexity of the class-based model

We then interpolated the class-based model with the best word-based model (Modified Kneser-Ney) resulting in the following model

$$p(w_i \mid w_{i-2}, w_{i-1}) = \lambda\, p(w_i \mid w_{i-2}, w_{i-1}) + (1-\lambda)\, p(g_i \mid g_{i-2}, g_{i-1})\, p(w_i \mid g_i) \quad (2)$$

Table 11 shows perplexity and word error rate of the interpolated model. We see no improvement over the plain Modified Kneser-Ney back-off model for the 1M and 5M training sets and a very small improvement for the 35M training set, which was statistically insignificant based on the NIST sign test [8].

	(1M)		(5M)		(35M)	
Model	PP	WER	PP	WER	PP	WER
Interp.	319	26.99	232	22.04	154	18.44

Table 11: Perplexity and word error rate of the interpolated model

A more interesting measure is the hit rate of the class-based model, which is shown in Table 12

	hit rate (%) 1M	hit rate (%) 5M	hit rate (%) 35M
1-gram	21.3	12.1	5.1
2-gram	56	50.4	37.6
3-gram	22.7	37.6	57.4

Table 12: Hit rate of 1-, 2-, 3-grams

Unfortunately, we have only small improvements on the hit rate of trigrams. Recall from Table 8 that the hit rates of the word trigram model were 20.2%, 33.7% and 52.6% for the three training sets. This means that, for example for the 35M training set, using classes based on stem we increased the

number of times the model uses a trigram probability only about 5%. We would expect a much larger percentage, given the inflectional nature of the Greek language. This percentage is even smaller for the 1M and 5M training sets (2.5% and 4% respectively), which explains the bad results on WER for these cases.

By examining the 35M training set, we found the number of unigrams and bigrams that became bigrams or trigrams in the class-based model (note: the class-based alone). Table 13 shows the results. The notation means that we had 193.545 unigrams in the word trigram model that remained unigrams in the class-based model, 82.399 unigrams that became bigrams etc.

11=193.545	12=82.399	13=7.651
21=0	22=1.356.024	23=176.151
31=0	32=0	33=2.013.034

Table 13: Number of n-grams that changed order between word-based and class-based model

Table 14 shows the perplexity of the word-based and class-based model for each subset. For example, the unigrams that remained unigrams had perplexity 64169 in the word-based model and 83327 in the class-based model. We see that for elements above the diagonal perplexity decreased because the order increased, but for elements on the diagonal perplexity increased.

11: 64169-83327	12: 31366-9055	13: 39147-1227
21: -	22: 542-901	23: 356-126
31: -	32: -	33: 27-41

Table 14: Difference in perplexity between word-based and class-based model

Table 15 shows the same information for the word-based and the interpolated model. We see that linear interpolation fails to successfully combine the different information sources. The interpolated model does not retain the improvement of the class-based model (the elements above the diagonal of Table 14). The perplexities of the interpolated model for each subset approximate that of the word-based model. This is due to the fact that in the interpolation the word-based model takes a high λ and the class-based model takes a small λ.

11: 64169-64944	12: 31366-22442	13: 39147-8866
21: -	22: 542-551	23: 356-268
31: -	32: -	33: 27-28

Table 15: Difference in perplexity between word-based and interpolated model

4. Maximum Entropy Models

Despite the insignificant improvement of the interpolated models, we were convinced that some information must exist in the stem of words. A better way to combine different information sources (n-gram and stem information) is via the maximum entropy method [9]. A conditional maximum entropy model has the following form

$$p(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(x, y) \right) \tag{3}$$

where $Z(x)$ is a normalizing constant

$$Z(x) = \sum_y \exp\left(\sum_i \lambda_i f_i(x, y) \right) \tag{4}$$

f_i are binary-valued feature functions and λ_i are some unknown constants, to be found. To search the exponential family defined by (3) for the unknown parameters λ_i, we use the Improved Iterative Scaling or IIS algorithm [9].

We built a maximum entropy model using unigrams, bigrams and trigram constraints. We used the Modified Kneser-Ney method to smooth the target feature expectations.

$$f(w) = f(x, y; w) = \begin{cases} 1 \text{ if } y = w \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

$$f(v, w) = f(x, y; v, w) = \begin{cases} 1 \text{ if } x \text{ ends in } v \text{ and } y = w \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

$$f(u, v, w) = f(x, y; u, v, w) = \begin{cases} 1 \text{ if } x \text{ ends in } u, v \text{ and } y = w \\ 0 \text{ otherwise} \end{cases} \tag{7}$$

This model has the following form

$$p(w \mid u, v) = \frac{e^{\lambda_w f(w)} e^{\lambda_{v,w} f(v,w)} e^{\lambda_{u,v,w} f(u,v,w)}}{Z(u, v)} \tag{8}$$

We then added to the model bigrams and trigrams constraints from the classification of words into classes based on stem

$$f(s(v), w) = f(x, y; s(v), w) = \begin{cases} 1 \text{ if } x = stem(v) \text{ and } y = w \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

$$f(s(u), s(v), w) = f(x, y; s(u), s(v), w) = \begin{cases} 1 \text{ if } x = stem(u), stem(v) \text{ and } y = w \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

resulting in the following model

$$p(w \mid u, v) = \frac{e^{\lambda_w f(w)} e^{\lambda_{v,w} f(v,w)} e^{\lambda_{u,v,w} f(u,v,w)}}{Z(u, v)} \times e^{\lambda_{s(v),w} f(s(v),w)} e^{\lambda_{s(u),s(v),w} f(s(u),s(v),w)} \tag{11}$$

Note that the denominator is $Z(u, v) = Z(u, v, s(u), s(v))$ because the stem of words is a one to one mapping. To use this model in speech recognition we used a mapping of model

parameters to ARPA back-off format similar to that proposed in [10]. Recall that, in a back-off trigram model

$$p(w \mid u,v) = \begin{cases} f(w \mid u,v) & \text{if } c(u,v,w) \geq threshold \\ bow(u,v)\,p(w \mid v) & \text{otherwise} \end{cases} \quad (12)$$

We map f and bow with the following quantities

$$\zeta = \sum_w e^{\lambda_w f(w)} \quad (13)$$

$$\zeta(v) = \sum_w e^{\lambda_w f(w)} e^{\lambda_{v,w} f(v,w)} e^{\lambda_{s(v),w} f(s(v),w)} \quad (14)$$

$$\zeta(u,v) = \sum_w e^{\lambda_w f(w)} e^{\lambda_{v,w} f(v,w)} e^{\lambda_{u,v,w} f(u,v,w)} \times e^{\lambda_{s(v),w} f(s(v),w)} e^{\lambda_{s(u),s(v),w} f(s(u),s(v),w)} \quad (15)$$

$$f(w \mid u,v) = \frac{e^{\lambda_w f(w)} e^{\lambda_{v,w} f(v,w)} e^{\lambda_{u,v,w} f(u,v,w)}}{\zeta(u,v)} \times e^{\lambda_{s(v),w} f(s(v),w)} e^{\lambda_{s(u),s(v),w} f(s(u),s(v),w)} \quad (16)$$

$$f(w \mid v) = \frac{e^{\lambda_w f(w)} e^{\lambda_{v,w} f(v,w)} e^{\lambda_{s(v),w} f(s(v),w)}}{\zeta(v)} \quad (17)$$

$$f(w) = \frac{e^{\lambda_w f(w)}}{\zeta} \quad (18)$$

$$bow(u,v) = \frac{\zeta(v)}{\zeta(u,v)} \quad (19)$$

$$bow(v) = \frac{\zeta}{\zeta(v)} \quad (20)$$

Table 16 shows the perplexity and word error rate results for the maximum entropy models incorporating n-gram and n-gram+stem constraints. We see an improvement in WER of 0.24%, 0.25% and 0.28% absolute for the three training sets respectively (compared with the Modified Kneser-Ney back-off model). After doing the NIST sign test, we found that all these improvements are statistically significant.

	(1M)		(5M)		(35M)	
Model	PP	WER	PP	WER	PP	WER
ME 3gram	331	26.83	239	21.94	158	18.6
ME 3gram +stem	320	26.54	227	21.66	143	18.29

Table 16: Perplexity and word error rate of maximum entropy models

Table 17 shows the perplexity of the word-based and maximum entropy stem-based model for the same subsets of words as in Tables 14 and 15.

11: 64169-65217	12: 31366-9832	13: 39147-1252
21: -	22: 542-565	23: 356-149
31: -	32: -	33: 27-30

Table 17: Difference in perplexity between word-based and maximum entropy model

We see that the maximum entropy model retains the improvements of both a class-based model alone and a word-based model. The maximum entropy model with stem constraints exhibits a back-off effect provided by the stem information, although there is no explicit notion of backing-off in maximum entropy models. To understand this, consider a trigram u, v, w not seen in the training data, with a stem trigram $stem(u), stem(v), w$ that was seen often. Then the trigram feature of Equation (7) would not be active, but the stem trigram feature of Equation (10) would be, which is equivalent as backing-off in a stem trigram (instead of backing-off in the bigram v, w of a standard n-gram back-off model). This is something in between a bigram and trigram classification of the history.

5. Conclusions

We presented a maximum entropy language model incorporating n-gram and stem constraints arisen from the classification of the words of an inflectional language like Greek into classes. We showed that using stem information, a small but statistically significant improvement in WER can be achieved.

6. References

[1] S. Young, M. Adda-Dekker, X. Aubert, C. Dugast, J.-L. Gauvain, D. Kershaw, L. Lamel, D. Leeuwen, D. Pye, A. Robinson, H. Steeneken, P. Woodland. "Multilingual large vocabulary speech recognition: the European SQALE project". *Computer Speech and Language, 1997.*

[2] S. Katz. "Estimation of probabilities from sparse data for the language model component of a speech recognizer". *IEEE Transactions on Acoustics, Speech and Signal Processing, 1987.*

[3] I. Good. "The population frequencies of species and the estimation of population parameters". *Biometrika, 1953.*

[4] I. Witten and T. Bell. "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression". *IEEE Transactions on Information Theory, 1991.*

[5] H. Ney, U. Essen and R. Kneser. "On structuring probabilistic dependencies in stochastic language modeling". *Computer Speech and Language, 1994.*

[6] S. Chen and J. Goodman. "An empirical study of smoothing techniques for language modeling". *In Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 1996.*

[7] V. Digalakis, D. Oikonomidis, D. Pratsolis, N.Tsourakis, C. Vosnidis, N. Chatzichrisafis, and V. Diakoloukas. "Large vocabulary continuous speech recognition in Greek: Corpus and an automatic dictation system". *Submitted to Eurospeech 2003.*

[8] http://www.nist.gov

[9] A. Berger, S. Della Pietra and V. Della Pietra. "A maximum entropy approach to natural language processing". *Computational Linguistics, 1996.*

[10] J. Wu and S. Khudanpur. "Building a topic-dependent maximum entropy language model for very large corpora". *In Proceedings of ICASSP, 2002*