Improved Modeling of Out-Of-Vocabulary Words Using Morphological Classes

Thomas Müller and Hinrich Schütze
Institute for Natural Language Processing
University of Stuttgart, Germany
muellets@ims.uni-stuttgart.de

Abstract

We present a class-based language model that clusters rare words of similar morphology together. The model improves the prediction of words after histories containing out-of-vocabulary words. The morphological features used are obtained without the use of labeled data. The perplexity improvement compared to a state of the art Kneser-Ney model is 4% overall and 81% on unknown histories.

1 Introduction

One of the challenges in statistical language modeling are words that appear in the recognition task at hand, but not in the training set, so called out-of-vocabulary (OOV) words. Especially for productive language it is often necessary to at least reduce the number of OOVs. We present a novel approach based on *morphological classes* to handling OOV words in language modeling for English. Previous work on morphological classes in English has not been able to show noticeable improvements in perplexity. In this article class-based language models as proposed by Brown et al. (1992) are used to tackle the problem. Our model improves perplexity of a Kneser-Ney (KN) model for English by 4%, the largest improvement of a state-of-the-art model for English due to morphological modeling that we are aware of. A class-based language model groups words into classes and replaces the word transition probability by a class transition probability and a word emission probability:

$$P(w_3|w_1w_2) = P(c_3|c_1c_2) \cdot P(w_3|c_3). \quad (1)$$

Brown et al. and many other authors primarily use context information for clustering. Niesler et al. (1998) showed that context clustering works better than clusters based on part-of-speech tags. However, since the context of an OOV word is unknown and it therefore cannot be assigned to a cluster, OOV words are as much a problem to a context-based class model as to a word model. That is why we use non-distributional features – features like morphological suffixes that only depend on the shape of the word itself – to design a new class-based model that can naturally integrate unknown words.

In related work, *factored language models* (Bilmes and Kirchhoff, 2003) were proposed to make use of morphological information in highly inflecting languages such as Finnish (Creutz et al., 2007), Turkish (Creutz et al., 2007; Yuret and Biçici, 2009) and Arabic (Creutz et al., 2007; Vergyri et al., 2004) or compounding languages like German (Berton et al., 1996). The main idea is to replace words by sequences of factors or features and to apply statistical language modeling to the resulting factor sequences. If, for example, words were segmented into morphemes, an unknown word would be split into an unseen sequence, which could be recognized using discounting techniques. However, if one morpheme, e.g. the stem, is unknown to the system, the fundamental problem remains unsolved.

Our class-based model uses a number of features that have not been used in factored models (e.g., shape and length features) and achieves – in contrast to factored models – good perplexity gains for English.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 524–528,
Portland, Oregon, June 19-24, 2011.

$is\_capital(w)$	first character of w is an uppercase letter
$is\_all\_capital(w)$	$\forall\, c \in w : c$ is an uppercase letter
$capital\_character(w)$	$\exists\, c \in w : c$ is an uppercase letter
$appears\_in\_lowercase(w)$	$\neg capital\_character(w) \vee w' \in \Sigma_T$
$special\_character(w)$	$\exists\, c \in w : c$ is not a letter or digit
$digit(w)$	$\exists\, c \in w : c$ is a digit
$is\_number(w)$	$w \in L([+-\epsilon][0-9]\,(([.,][0-9])\|[0-9])*)$
$not\_special(w)$	$\neg(special\_character(w) \vee digit(w) \vee is\_number(w))$

Table 1: Predicates of the capitalization and special character groups. Σ_T is the vocabulary of the training corpus T, w' is obtained from w by changing all uppercase letters to lowercase and $L(expr)$ is the language generated by the regular expression $expr$.

2 Morphological Features

The feature vector of a word consists of four parts that represent information about *suffixes*, *capitalization*, *special characters* and *word length*. For the suffix group, we define a binary feature for each of the 100 most frequent suffixes learned on the training corpus by the Reports algorithm (Keshava, 2006), a general purpose unsupervised morphology learning algorithm. One additional binary feature is used for all other suffixes learned by Reports, including the empty suffix.

The feature groups *capitalization* and *special characters* are motivated by the analysis shown in Table 2. Our goal is to improve OOV modeling. The table shows that most OOV words ($f = 0$) are numbers (CD), names (NP), and nouns and adjectives (NN, NNS, JJ). This distribution is similar to hapax legomena ($f = 1$), but different from the POS distribution of all tokens. Capitalization and special character features are of obvious utility in identifying the POS classes NP and CD since names in English are usually capitalized and numbers are written with digits and special characters such as comma and period. To capture these "shape" properties of a word, we define the features listed in Table 1.

The fourth feature group is length. Short words often have unusual distributional properties. Examples are abbreviations and bond credit ratings like Aaa. To represent this information in the *length* part of the vector, we define four binary features for lengths 1, 2, 3 and greater than 3. The four parts of the vector (suffixes, capitalization, special characters, length) are weighted equally by normalizing the subvector of each subgroup to unit length.

We designed the four feature groups to group word types to either resemble POS classes or to induce an even finer sub-partitioning. Unsupervised POS clustering is a hard task in English and it is virtually impossible if a word's context (which is not available for OOV items) is not taken into account. For example, there is no way we can learn that "the" and "a" are similar or that "child" has the same relationship to "children" as "kid" does to "kids". But as our analysis in Table 2 shows, part of the benefit of morphological analysis for OOVs comes from an appropriate treatment of names and numbers. The suffix feature group is useful for categorizing OOV nouns and adjectives because there are very few irregular morphemes like "ren" in *children* in English and OOV words are likely to be regular words.

So even though morphological learning based on the limited information we use is not possible in general, it can be partially solved for the special case of OOV words. Our experimental results in Section 5 confirm that this is the case. We also testes prefixes and features based on word stems. However, they produced inferior clustering solutions.

3 The Language Model

As mentioned before in the literature, e.g. by Maltese and Mancini (1992), class-based models only outperform word models in cases of insufficient data. That is why we use a frequency-based approach and only include words below a certain token frequency threshold θ in the clustering process. A second motivation is that the contexts of low frequency words are more similar to the expected contexts of OOV words.

Given a training corpus, all words with a fre-

tag	types		tokens
	$f = 1$	$f = 0$ (OOV)	
CD	0.39	0.38	0.05
NP	0.35	0.35	0.14
NN	0.10	0.10	0.17
NNS	0.05	0.06	0.07
JJ	0.05	0.06	0.07
V*	0.04	0.05	0.15
Σ	0.98	0.99	0.66

Table 2: Proportion of dominant POS for types with training set frequencies $f \in \{0, 1\}$ and for tokens. V* consists of all verb POS tags.

quency below the threshold θ are partitioned into k clusters using the bisecting k-means algorithm (Steinbach et al., 2000). The cluster of an OOV word w can be defined as the cluster whose centroid is closest to the feature vector of w. The formerly removed high-frequency words are added as singleton clusters to produce a complete clustering. However, OOV words can only be assigned to the original k-means clusters. Over this clustering a class-based trigram model can be defined, as introduced by Brown et al. (1992). The word transition probability of such a model is given by equation 1, where c_i denotes the cluster of the word w_i. The class transition probability $P(c_3|c_1c_2)$ is estimated using the unsmoothed maximum likelihood estimate. The emission probability is defined as follows:

$$P(w_3|c_3) = \begin{cases} 1 & \text{if } c(w_3) > \theta \\ (1-\epsilon)\frac{c(w_3)}{\sum_{w \in c_3} c(w)} & \text{if } \theta \geq c(w_3) > 0 \\ \epsilon & \text{if } c(w_3) = 0 \end{cases}$$

where $c(w)$ is the frequency of w in the training set.

ϵ is estimated on held-out data. The morphological language model is then interpolated with a modified Kneser-Ney trigram model. In this interpolation the parameters λ depend on the cluster c_2 of the history word w_2, i.e.:

$$P(w_3|w_1w_2) = \lambda(c_2) \cdot P_M(w_3|w_1w_2) + (1 - \lambda(c_2)) \cdot P_{KN}(w_3|w_1w_2).$$

This setup may cause overfitting as every high frequent word w_2 corresponds to a singleton class. A grouping of several words into equivalence classes could therefore further improve the model; this, however, is beyond the scope of this article. We estimate optimal parameters $\lambda(c_2)$ using the algorithm described by Bahl et al. (1991).

4 Experimental Setup

We compare the performance of the described model with a Kneser-Ney model and an interpolated model based on part-of-speech (POS) tags. The relation between words and POS tags is many-to-many, but we transform it to a many-to-one relation by labeling every word – independent of its context – with its most frequent tag. OOV words are treated equally even though their POS classes would not be known in a real application. Treetagger (Schmid, 1994) was used to tag the entire corpus.

The experiments are carried out on a Wall Street Journal (WSJ) corpus of 50 million words that is split into training set (80%), valdev (5%), valtst (5%), and test set (10%). The number of distinct feature vectors in training set, valdev and validation set (valdev+valtst) are 632, 466, and 512, respectively. As mentioned above, the training set is used to learn suffixes and the maximum likelihood n-gram estimates. The unknown word rate of the validation set is $\epsilon \approx 0.028$.

We use two setups to evaluate our methods. The first uses *valdev* for parameter estimation and *valtst* for testing and the second the entire validation set for parameter estimation and the test set for testing. All models with a threshold greater or equal to the frequency of the most frequent word type are identical. We use ∞ as the threshold to refer to these models. In a similar manner, the cluster count ∞ denotes a clustering where two words are in the same cluster if and only if their features are identical. This is the finest possible clustering of the feature vectors.

5 Results

Table 3 shows the results of our experiments. The KN model yields a perplexity of 88.06 on *valtst* (top row). For small frequency thresholds overfitting effects cause that the interpolated models are worse than the KN model. We can see that a clustering of the feature vectors is not necessary as the differences between all cluster models are small and c_∞ is the overall best model. Surprisingly, morphological clustering and POS classes are close even though

θ	c_{POS}	c_1	c_{50}	c_{100}	c_∞
0	88.06	88.06	88.06	88.06	88.06
1	89.74	89.84	89.73	89.74	89.74
5	89.07	89.36	89.07	89.06	89.07
10	88.59	89.01	88.58	88.57	88.58
50	86.72	87.58	86.69	86.68	86.68
10^2	85.92	87.06	85.92	85.91	85.89
10^3	**84.43**	86.88	84.83	84.77	**84.56**
10^4	85.22	87.59	85.89	85.73	85.26
10^5	86.82	87.99	87.44	87.32	86.79
∞	87.31	88.06	87.96	87.92	87.62

θ	c_{POS}	c_1	c_{50}	c_{100}	c_∞
0	813.50	813.50	813.50	813.50	813.50
1	181.25	206.17	182.78	183.62	184.43
5	152.51	185.54	154.52	152.98	153.83
10	147.48	186.12	149.34	147.98	147.48
50	**146.21**	203.10	142.21	140.67	**140.46**
10^2	149.06	215.54	143.95	142.48	141.67
10^3	173.91	279.02	164.22	159.04	150.13
10^4	239.72	349.54	221.42	208.85	180.57
10^5	317.13	373.98	318.04	297.18	236.90
∞	348.76	378.38	366.92	357.80	292.34

Table 3: Perplexities for different frequency thresholds θ and cluster models. In the left table, perplexity is calculated over all events $P(w_3|w_1w_2)$ of the *valtst* set. On the right side, the subset of events where w_1 or w_2 are unknown is taken into account. The overall best results for class models and POS models are highlighted in bold.

the POS class model uses oracle information to assign the right POS to an unknown word. The optimal threshold is $\theta = 10^3$ – the bolded perplexity values 84.43 and 84.56; that means that only 1.35% of the word types were excluded from the morphological clustering (86% of the tokens). The improvement over the KN model is 4%.

In a second evaluation we reduce the perplexity calculations to predictions of the form $P(w_3|w_1w_2)$ where w_1 or w_2 are OOV words. On such an event the KN model has to back off to a bigram or even unigram estimate, which results in inferior predictions and higher perplexity. The perplexity for the KN model is 813.50 (top row). A first observation is that the perplexity of model c_1 starts at a good value, but worsens with rising values for $\theta \geq 10$. The reason is the dominance of proper nouns and cardinal numbers at a frequency threshold of one and in the distribution of OOV words (cf. Table 2). The c_1 model with $\theta = 1$ is specialized for predicting words after unknown nouns and cardinal numbers and two thirds of the unknown words are of exactly that type. However, with rising θ, other word classes get a higher influence and different probability distributions are superimposed. The best morphological model c_∞ reduces the KN perplexity of 813.50 to 140.46 (bolded), an improvement of 83%.

As a final experiment, we evaluated our method on the test set. In this case, we used the entire validation set for parameter tuning (i.e., valdev and valtst). The overall perplexity of the KN model is 88.28, the perplexities for the best POS and c_∞ cluster model for $\theta = 1000$ are 84.59 and 84.71 respectively, which corresponds again to an improvement of 4%. For unknown histories the KN model perplexity is 767.25 and the POS and c_∞ cluster model perplexities at $\theta = 50$ are 150.90 and 144.77. Thus, the morphological model reduces perplexity by 81% compared to the KN model.

6 Conclusion

We have presented a new class-based morphological language model. In an experiment the model outperformed a modified Kneser-Ney model, especially in the prediction of the continuations of histories containing OOV words. The model is entirely unsupervised, but works as well as a model using part-of-speech information.

Future Work. We plan to use our model for domain adaptation in applications like machine translation. We then want to extend our model to other languages, which could be more challenging, as certain languages have a more complex morphology than English, but also worthwhile, if the unknown word rate is higher. Preliminary experiments on German and Finnish show promising results. The model could be further improved by using contextual information for the word clustering and training a classifier based on morphological features to assign OOV words to these clusters.

Acknowledgments. This research was funded by DFG (grant SFB 732). We would like to thank Helmut Schmid and the anonymous reviewers for their valuable comments.

References

Lalit R. Bahl, Peter F. Brown, Peter V. de Souza, Robert L. Mercer, and David Nahamoo. 1991. A fast algorithm for deleted interpolation. In *Speech Communication and Technology*, pages 1209–1212.

Andre Berton, Pablo Fetter, and Peter Regel-Brietzmann. 1996. Compound words in large-vocabulary German speech recognition systems. In *Spoken Language*, volume 2, pages 1165 –1168 vol.2, October.

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Human Language Technology*, NAACL '03, pages 4–6. Association for Computational Linguistics.

Peter F. Brown, Peter V. de Souza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, December.

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5:3:1–3:29, December.

Samarth Keshava. 2006. A simpler, intuitive approach to morpheme induction. In *PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, pages 31–35.

Giulio Maltese and Federico Mancini. 1992. An automatic technique to include grammatical and morphological information in a trigram-based statistical language model. In *Acoustics, Speech, and Signal Processing*, volume 1, pages 157 –160 vol.1, March.

Thomas R. Niesler, Edward W.D. Whittaker, and Philip C. Woodland. 1998. Comparison of part-of-speech and automatically derived category-based language models for speech recognition. In *Acoustics, Speech and Signal Processing*, volume 1, pages 177 –180 vol.1, May.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *New Methods in Language Processing*, pages 44–49.

Michael Steinbach, George Karypis, and Vipin Kumar. 2000. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*.

Dimitra Vergyri, Katrin Kirchhoff, Kevin Duh, and Andreas Stolcke. 2004. Morphology-based language modeling for Arabic speech recognition. In *Spoken Language Processing*, pages 2245–2248.

Deniz Yuret and Ergun Biçici. 2009. Modeling morphologically rich languages using split words and unstructured dependencies. In *International Joint Conference on Natural Language Processing*, pages 345–348. Association for Computational Linguistics.