Clustered Language Models based on Regular Expressions for SMT

Saša Hasan and Hermann Ney

Lehrstuhl für Informatik VI – Computer Science Department
RWTH Aachen University
52056 Aachen, Germany
{hasan,ney}@cs.rwth-aachen.de

Abstract. In this paper, we present a language model based on clusters obtained by applying regular expressions to the training data and, thus, discriminating several different sentence types as, e.g. interrogatives, imperatives or enumerations. The main motivation lies in the observation that different sentence types also underlie a different syntactic structure, and thus yield a varying distribution of n-grams reflecting their word order. We show that this assumption is valid by applying the models to English-Spanish bilingual corpora and obtaining good perplexity reductions of approximately 25%. In addition, we perform an n-best rescoring experiment and show a relative improvement of 4-5% in word error rate. The models can be easily adapted to other translation tasks and do not need complicated training methods, thus being a valuable alternative for on-demand rescoring of sentence hypotheses such as they occur in the CAT framework.

1 Introduction

Language modeling is a rather long-established research field in the area of Natural Language Processing. In Automatic Speech Recognition (ASR), the language model guides the acoustic analysis by specifying the order in which a sequence of words is likely to occur. In Statistical Machine Translation (SMT), where the best translation $\hat{e}_1^I$ of source words f_1^J is obtained by maximizing the conditional probability

$$\begin{aligned}\hat{e}_1^I &= \operatorname*{argmax}_{e_1^I}\{Pr(e_1^I|f_1^J)\} \\ &= \operatorname*{argmax}_{e_1^I}\{Pr(f_1^J|e_1^I)\cdot Pr(e_1^I)\}\end{aligned} \tag{1}$$

by using Bayes decision rule, the first probability on the right-hand side of the equation denotes the translation model whereas the second is the language model of the target language. The prevalent approach of modeling $Pr(e_1^I)$ is based on n-grams with suitable smoothing methods:

$$Pr(e_1^I) = \prod_{i=1}^{I} Pr(e_i|e_{i-n+1}^{i-1}) \tag{2}$$

Due to the problem of sparse data, high order n-grams, i.e. where $n > 3$, are rarely applied. Thus, only local syntactic dependencies are captured in a conventional trigram or bigram.

In the past, additional models have been proposed that boost the performance of simple trigram models. It is shown in (Martin et al., 1999; Goodman, 2000) that a combination of individual techniques based on caching, skipping, clustering or sentence mixtures improves the baseline significantly. In this approach, we will present a clustering technique that is based on regular expressions. The motivation behind this lies in the following observation: the syntactic structure of a sentence is influenced by its type. It is obvious that an interrogative sentence has a different structure from a declarative one due to non-local dependencies arising e.g. from *wh*-extraction. As an example, consider the syntax of the following sentences:

- *What are distribution templates?*
- *Distribution templates are what were previously referred to as templates or scan templates.*

If we look closer at the first four words of each sentence (*what*, *are*, *distribution* and *templates*), the trigrams observed are quite different, leading to the hypothesis that a language model that can discrim-

inate between these cases also performs better than the traditional approach.

The method that we apply in order to cluster the sentences into specific classes is based on regular expressions. A very simple trigger for an interrogative class is e.g. a question mark "?". This information is then used to train class-specific language models which are interpolated with the main language model in order to elude data sparseness. A possible practical application of this method is in the area of Computer Aided Translation (CAT) where an MT system usually provides a list of sentence hypotheses to the translator. This list can be reordered on demand by applying additional rescoring steps which use the presented language model.

In Section 2, we describe the framework of the language model based on clusters obtained from applying regular expressions to a training corpus. Section 3 reports experimental results on several corpora in terms of perplexity reduction and word error rate by using an n-best list rescoring framework. The results are discussed and an overview on future work is given in Section 4.

2 Framework

The conventional way of using sentence-level mixture models (as e.g. in (Iyer and Ostendorf, 1999)) is to calculate the overall probability of a sentence w_1^N as

$$Pr(w_1^N) = \sum_{c=1}^{C} \lambda_c \left(\prod_{n=1}^{N} Pr_c(w_n|w_{n-2}^{n-1}) \right), \quad (3)$$

where C is the number of classes (or "topics"), $Pr_c(\cdot|\cdot)$ denotes the class-dependent trigram probability and the λ_c's are the sentence-level mixture weights. Usually, this model is also linearly interpolated with a global language model trained on all data, since the partitioning into class-dependent subsets reduces the available training material within one class. One possible disadvantage of this approach is that the mixture weights are determined globally on the whole data, i.e. all classes have a smoothed influence on the test data, although each sentence probably belongs only to one class.

As an alternative, we propose the following approach: instead of a mixture model on all classes, we use a trigger-based model that combines only two models at a time, namely the class-specific model corresponding to the matching regular expression (RE) and the global language model. For a sentence w_1^N whose matching class $\mathrm{RE}(w_1^N) = c$, we obtain the probability

$$\begin{aligned} Pr(w_1^N) &= \lambda_c \prod_{n=1}^{N} Pr_c(w_n|w_{n-2}^{n-1}) + \\ &\quad (1-\lambda_c) \prod_{n=1}^{N} Pr_g(w_n|w_{n-2}^{n-1}), \quad (4) \end{aligned}$$

where $Pr_g(\cdot|\cdot)$ is the global model and λ_c is set to zero in case of no matching regular expressions, so we back-off to the global model. Ideally, the sentences falling into one class share a similar upper-level syntactic structure. Another advantage of this approach is that this kind of clustering groups sentences with similar words such as e.g. *wh*-words and therefore also the same set of related words occurring in interrogative sentence types. Thus, an additional unigram cache is added to the global model with a small weight. Results indicating that a combination of the class-specific model and the unigram cache model is fruitful are reported in Section 3.2.

Since the interpolation only takes two models at a time, no complex re-estimation techniques of the weights λ_c are necessary. A simple hill-climbing algorithm quickly finds the global maximum on the interpolated graph for log-likelihood from held-out data (development set). Another interesting feature of the proposed model is that, during training, sentences are reused if matching several regular expressions, which has a positive effect on the overall size of the training data. In testing, only the first matching regular expression is applied. The next section describes the experiments and also gives an overview on perplexity results, training sizes and individual improvements for specific triggers.

3 Experiments

The experiments are set in the machine translation area and are focused on two aspects. Firstly, we want to use the notion of *perplexity* as an evaluation criterion. It denotes the inverse geometric average of the branching factor after each word. So for a

sentence w_1^N, we obtain the perplexity by calculating

$$\mathrm{PP} = [Pr(w_1^N)]^{-1/N}. \quad (5)$$

The higher the perplexity, the more difficult the task, since the system has more competing candidates to choose from at each position. It has been shown that perplexity reduction is correlated with reductions in error rate (Klakow and Peters, 2002). As a rule of thumb, (Rosenfeld, 2000) notes that 10-20% reductions are noteworthy and usually result in some improvement, whereas 30% or more over a good baseline is quite significant.[1] Thus, we additionally carry out a rescoring experiment using n-best lists generated from a word graph to check this claim. Secondly, we take a closer look at what kind of triggers achieve what kind of reduction, in order to conclude which triggers are useful and which are not.

3.1 Corpora

The investigated corpora are the simplified English-Spanish Xerox corpus (technical manuals for printing devices) for general performance of the trigger approach, and the English-Spanish LC-STAR corpus (dialogues in the domain of appointment scheduling and travel planning) for specific triggers based on verb POS-tag information. The corpus statistics are summarized in Table 1.

For the Xerox corpus, 9 triggers have been selected which try to reflect the basic structure/type of a sentence. Since the corpus is a technical manual, the sentences are rather short and there are also a lot of enumerations and elliptical clauses as they often appear in "navigation" dialogues, e.g. "*canceling a scheduled operation*". In this case, one of the possible triggers is the regular expression `[^.!?]$` which matches all sentences that do not end in a common punctuation mark and where we therefore can expect a special structural property of e.g. a missing verbal phrase. Table 2 lists the nine triggers used in the experiments, together with the number of matching sentences in training, development (both where a sentence can match multiple times) and testing, where the matches are prioritized, i.e. the model of the first matching regular expression is applied.

[1] These values are known from experience.

As can be seen in the table, 1006 test sentences are covered by class-specific triggers, whereas the remaining part (119 sentences) is backed off entirely to the global model. The corpus is a simplified version of the raw format. The preprocessing step involves the conversion of all words to lower case, tokenization (i.e. splitting of punctuation marks, parentheses, etc. from the words) and categorization, i.e. many tokens (especially numbers, special characters like parentheses, bullet markers such as "$*$" or "a)", etc.) are replaced by special ones, which basically reduces the overall vocabulary size and, thus, the perplexity of the models. We will show results of additional experiments with the *raw* version of the corpus (together with non-simplified versions of the corpora for the language pairs English-French and English-German) in Section 3.3.

For the LC-STAR corpus, part-of-speech tagged data is provided. So the second trigger-based approach was to classify the sentences according to their *verb* POS-tag information, since the verb is usually regarded as the head of the sentence, influencing most of its syntactic structure. Given that Spanish is much more inflectional than English, we set Spanish as the primary target language for this corpus and extracted the most frequent Spanish verb POS-tag combinations together with 3 additional triggers, namely for interrogatives, exclamations and sentences with no verb POS-tags (ellipsis), resulting in a total of 36 class-specific regular expressions. For this setting, the total number of matches in the training data was 77884 (cf. Table 2), which is almost double the amount of the initially available data, thus reducing the overall data scarcity of the clustered models. This means that each matched sentence contributes to approximately two clusters on average during training which has a positive effect on the vocabulary of the class-specific models. For the development section (which is used for the estimation of the class-specific mixture weights), the total number of matches was 2163 (in contrast to the initial 972 sentences).

3.2 Results

This section presents the reductions in perplexity as well as word error rates for the given corpora. The

		Xerox		LC-STAR	
		Spanish	English	Spanish	English
TRAIN	Sentences	55761		40574	
	Running Words (with punc. marks)	752606	665399	516717	482290
	Vocabulary	11050	7956	8116	14327
	Singletons	3156	1928	3081	6743
DEV	Sentences	1012		972	
	Running Words (with punc. marks)	15957	14278	13983	12883
	Vocabulary	1433	1224	1584	1988
	OOVs (running words)	54	27	100	214
	OOVs (in voc.)	43	19	95	209
TEST	Sentences	1125		972	
	Running Words (with punc. marks)	10106	8370	13922	12771
	Vocabulary	1215	1132	1583	1997
	OOVs (running words)	69	49	124	213
	OOVs (in voc.)	39	26	117	206

Table 1. Corpus statistics for Spanish-English: Xerox (simplified) and LC-STAR.

Xerox	number of matches		
RE trigger	#train	#dev	#test
_QUESTION	271	5	4
_QUOTE	1264	6	9
_BRACKET	4722	107	52
_BULLET	7648	311	115
_SLASH	3682	71	31
_NUM	7572	78	35
:	7277	127	57
[^.!?]$	18977	252	677
_OTHERS	10222	124	26
total matched	61635	1081	1006
not matched	19776	307	119
ratio matches/sent.	1.11	1.07	0.89

LC-STAR	number of matches		
RE trigger	#train	#dev	#test
^([^/][^V])+$	1325	25	21
!	2479	92	64
^.*VMIF1S0.*$	968	10	4
\?	8319	204	44
^.*VSIP3P0.*$	710	26	14
^.*VMIP3P0.*$	1776	58	26
^.*VMIP2S0.*$	979	6	6
^.*VMIF1P0.*$	637	15	7
all remaining REs	60691	1727	635
total matched	77884	2163	821
not matched	8126	148	151
ratio matches/sent.	1.92	2.23	0.85

Table 2. Regular expression triggers used for the simplified Xerox technical manuals and LC-STAR corpus, and their corresponding number of matches in training, development and test data.

general observation is that clusters reflecting interrogatives, exclamations and elliptical constructs (i.e. sentences without a verbal phrase) achieve the highest perplexity reductions. So the approach described in Section 2 works especially well for these types. The best class-specific reductions for both corpora are listed in Table 3.

For the Xerox corpus, the perplexity results for both languages, English and Spanish, are shown in Table 4. Here, a significant improvement for both the class-specific as well as the unigram cache model can be observed. Since the data are technical manuals, terms like e.g. *printer* or *network* occur quite often and explain the good performance of the cache model. Additionally, the combination of both models even outperforms each of the individual approaches by far. Two basic language models are taken for comparison. The first one is a sim-

Xerox	Perplexity reduction		
English	5grKN	+mix	rel.imp.
_QUESTION	9.5	5.8	39.3%
[^.!?]$	28.4	21.0	26.1%
:	45.9	35.7	22.3%
_BULLET	33.4	27.8	16.8%
_BRACKET	63.9	56.1	12.2%
_OTHERS	15.6	13.7	12.0%
_QUOTE	40.9	37.0	9.6%
Spanish			
_QUESTION	10.9	6.5	40.5%
[^.!?]$	17.4	13.9	20.0%
:	39.6	33.2	16.1%
_QUOTE	49.0	41.7	15.0%
_BULLET	22.9	19.7	13.9%
_BRACKET	49.9	44.6	10.7%
_NUM	22.2	20.5	7.7%

LC-STAR Spanish	3grGT	+mix	rel.imp.
^([^/][^V])+$	23.1	12.1	47.8%
!	7.3	4.0	46.0%
^.*VMIF1S0.*$	96.0	67.3	29.9%
\?	26.4	20.8	21.0%
^.*VSIP3P0.*$	98.1	78.4	20.1%
^.*VMIP3P0.*$	67.7	54.2	19.9%
^.*VMIP2S0.*$	139.5	116.7	16.4%
^.*VMIF1P0.*$	46.1	39.4	14.4%

Table 3. Best performing regular expressions (in terms of relative perplexity reduction) for the class-specific language model for both tested corpora (using a KN-discounted 5-gram with cache for the Xerox task and a standard GT-discounted trigram for LC-STAR).

ple trigram using Katz back-off and Good-Turing discounting. This setting is the most used throughout the language modeling community, since it is fast to train and the common baseline of language modeling toolkits such as SRILM (Stolcke, 2002). The second approach is an advanced 5-gram which uses modified Kneser-Ney discounting (Goodman and Chen, 1998). As we can see, the baseline of the 5-gram for English (39.0) is almost 20% better than the simple trigram approach (48.3). The class-specific mixture based on clusters, which is obtained by applying regular expressions and utilizing a unigram cache, outperforms this baseline by an additional 25%. For the Spanish part, the behavior is similar. The trigram baseline is lowered from 32.9 to 24.4 (25% relative improvement), whereas the class-specific 5-gram approach yields an additional 19% relative reduction from 25.2 to 20.3.

Xerox	English		Spanish	
	PPL	rel.imp.	PPL	rel.imp.
3gramGT	48.3		32.9	
+mix	36.3	24.8%	26.4	19.8%
+cache	37.9	21.5%	28.6	13.1%
+mix+cache	32.2	33.3%	24.4	25.8%
5gramKN	39.0	*(19.3%)*	25.2	*(23.4%)*
+mix	32.1	17.7%	21.6	14.3%
+cache	32.6	16.4%	22.4	11.1%
+mix+cache	29.2	25.1%	20.3	19.4%

Table 4. Perplexity results on the Xerox corpus by comparing a traditional Katz back-off trigram model with Good-Turing discounting and a modified Kneser-Ney discounted 5-gram. The parenthesized numbers denote the relative improvement on the trigram baseline, "+mix" is the class-specific LM based on regular expressions, "+cache" the unigram cache model.

In order to see if these results can be directly used in a statistical machine translation framework, we carried out rescoring experiments based on n-best lists generated with our phrase-based state-of-the-art machine translation system (Bender et al., 2004). After training and optimization of all model scaling factors on the development n-best list with $n = 10\,000$, we extract all target sentence hypotheses over the whole list and match them to the regular expressions. For each cluster, its class-specific language model is applied and the costs (i.e. negative log-likelihoods) are added to the initial models of the original n-best list. We use the best model from the previous section, i.e. the class-specific Kneser-Ney smoothed 5-gram, but without the cache component, since the numerous hypotheses of a sentence do not differ much and, thus, the cache component does not help to discriminate between the various targets. The scaling factors are again optimized on the development list via the Downhill-Simplex algorithm. They are then taken in order to extract the best hypothesis from each source sentence of the test list. The results of this rescoring step are summarized in Table 5. The oracle-best error rates (WER/PER)

Spanish → English	WER[%]	PER[%]	BLEU[%]	NIST
10 000-best baseline	29.2	19.8	64.1	8.83
+ class-specific LM rescoring	28.1	19.1	65.2	8.90
English → Spanish				
10 000-best baseline	26.5	19.1	70.2	9.36
+ class-specific LM rescoring	25.2	18.1	72.0	9.40

Table 5. Translation results using the class-specific mixture LM with 5-grams on 10000-best lists of the Xerox corpus.

for the Spanish-English and English-Spanish n-best list are 14.9%/12.4% and 14.4%/12.0%, respectively, i.e. the error rates of the best hypotheses compared to the reference translations are half of the baseline error rate of the system. The results are consistent with those already observed for the perplexities. As can be seen, the word error rate (WER) decreases 1.1% absolute for English and 1.3% absolute for Spanish as target language. We also find relative improvements of 2-3% in BLEU scores.

For the experiment using the POS-tag information, the following regular expressions can be found among the best performing ones:

- *1st pers. sing., future tense* (`VMIF1S0`),
- *3rd pers. pl., present tense of* ser *(to be)* (`VSIP3P0`),
- *3rd pers. pl., present tense* (`VMIP3P0`),
- *2nd pers. sing., present tense* (`VMIP2S0`),
- *1st pers. pl., future tense* (`VMIF1P0`).

From this, one can conclude that, for the given domain, the subject number and person, as well as tense and modality information play an important role for the overall structure of the sentence. Although there were individual classes that performed well (cf. Table 3), the overall perplexity reduction for the Spanish portion of the LC-STAR corpus was only from 48.2 to 42.9 (11% relative) for the standard trigram and some additional 6% down to 40.2 when using a KN-discounted 5-gram.

A rescoring experiment was carried out and did not show significant improvements in terms of error rate (0.1% for WER and PER, 0.4% for BLEU). This can also originate from the poor quality of the POS tagger which was applied to all Spanish hypotheses in the n-best list. The small improvement is also due to the fact that the unigram cache did not significantly help when combined with the class-specific mixture model. A possible explanation for this is the "inconsistency" of the test sentences which seem to be chosen at random from the corpus and, thus, do not constitute chunks from consecutive dialogs.

3.3 Additional experiments

We also performed additional experiments using the raw versions (tokenized with normal case information (i.e. no lowercasing is applied), but not categorized) of the Xerox corpus for English, Spanish, French and German. Since the corpora differ for each language pair (English-Spanish, English-French and English-German), we also obtain three different perplexities for English. Table 6 gives an overview of the result. The triggers are basically modeled after the ones for the simplified corpus but are more fine-grained because of the missing categorization. So, e.g., the `_BULLET` trigger is replaced by three separate regular expressions that match a sentence if tokens are identified that mark the beginning of an ordered list: `^[0-9]+` (e.g. "1", "2", "3"), `^[a-z]\)` (e.g. "a)", "b)", "c)") and `^\*` (normal bullet "$*$").

The last experiment conducted was to test the regular expressions that worked best in the previous experiments on a large corpus. We used parts of the Wall Street Journal (all articles from 87-89) comprising of approximately 40 million running words of training data (without the set-aside articles for development and test) and applied the clustered language model using three classes, namely for interrogatives, exclamations and ellipsis (assumed if no

Xerox (raw)	5grKN	+mix	rel.imp.
English	76.8	54.8	28.7%
Spanish	42.4	33.4	21.2%
English	89.4	68.6	23.3%
French	63.7	52.0	18.4%
English	50.0	44.3	11.4%
German	85.8	72.5	15.5%

WSJ	3grGT	+mix	rel.imp.
(only match.)	155.6	133.3	14.3%

Table 6. Additional perplexity results on the raw Xerox corpus for different language pairs (English-Spanish, English-French and English-German) and the matched parts of the WSJ.

period is present at the end of the sentence). Although moderate perplexity reductions within the matched classes were achieved (cf. Table 6), the overall reduction on the whole test set was only 1.4%, since more than 90% of the corpus did not fall into any of the classes and was entirely backed off to the global model. Additional experiments have to be carried out in order to find more useful sentence types that can be identified by (probably more complex) regular expressions.

4 Conclusion

In this paper, we presented a new clustered language model that is based on applying regular expressions to the training data in order to train sentence-type class-specific language models. Each matching model is interpolated with a global model, which again uses a unigram cache component. The preliminary results look promising in terms of perplexity reduction, as well as error rates obtained for a translation task using an n-best list rescoring framework. Future translation experiments will include additional language pairs, such as English-German and English-French, as well as a closer look at the performance of other regular expression triggers. Here, we only present simple upper-level triggers, but regular expressions in general can model much more structural properties. So it is thinkable to conduct more experiments in this direction.

A possible drawback is that, currently, we look into the (development) data and select good triggers manually (though we presented a list of regular expressions that seem to work reliably in general, namely triggers that detect interrogative sentences, exclamations and ellipsis within phrases). As an extension, clustering techniques which are capable of finding the optimal set of clusters and methods that automatically derive promising triggers, are to be investigated. Since a sentence can be matched by more than one regular expression in training, we also observe an increase in the effective data size used for the class-specific models. Therefore, the problem arising from data sparseness for the class-specific models is reduced.

Acknowledgement

This work has been partly funded by the European Union under the RTD project TransType2 (IST-2001-32091), the integrated project TC-STAR – Technology and Corpora for Speech to Speech Translation – (IST-2002-FP6-506738) and by the Deutsche Forschungsgemeinschaft (DFG) under the project “Statistische Textübersetzung” (Ne572/5).

5 References

Oliver Bender, Richard Zens, Evgeny Matusov, and Hermann Ney. 2004. Alignment Templates: the RWTH SMT System. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pages 79–84, Kyoto, Japan, September.

Joshua Goodman and Stanley F. Chen. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, August.

Joshua T. Goodman. 2000. Putting it all together: Language model combination. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey.

Rukmini M. Iyer and Mari Ostendorf. 1999. Modeling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39, January.

Dietrich Klakow and Jochen Peters. 2002. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1):19–28.

Sven Martin, Christoph Hamacher, Jörg Liermann, Frank Wessel, and Hermann Ney. 1999. Assessment of smoothing methods and complex stochastic language modeling. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 1939–1942, Budapest, Hungary, September.

Ronald Rosenfeld. 2000. Two decades of statistical language modeling: where do we go from here? *Proceedings of IEEE*, 88(8):1270–8.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September.