Improving cross-domain n-gram language modelling with skipgrams

Louis Onrust
CLS, Radboud University Nijmegen
ESAT-PSI, KU Leuven
l.onrust@let.ru.nl

Antal van den Bosch
CLS, Radboud University Nijmegen
a.vandenbosch@let.ru.nl

Hugo Van hamme
ESAT-PSI, KU Leuven
hugo.vanhamme@esat.kuleuven.be

Abstract

In this paper we improve over the hierarchical Pitman-Yor processes language model in a cross-domain setting by adding skipgrams as features. We find that adding skipgram features reduces the perplexity. This reduction is substantial when models are trained on a generic corpus and tested on domain-specific corpora. We also find that within-domain testing and cross-domain testing require different backoff strategies. We observe a 30-40% reduction in perplexity in a cross-domain language modelling task, and up to 6% reduction in a within-domain experiment, for both English and Flemish-Dutch.

1 Introduction

Since the seminal paper on hierarchical Bayesian language models based on Pitman-Yor processes (Teh, 2006), Bayesian language modelling has regained an interest. Although Bayesian language models are not new (MacKay and Peto, 1995), previously proposed models were reported to be inferior compared to other smoothing methods. Teh's work was the first to report on improvements over interpolated Kneser-Ney smoothing (Teh, 2006).

To overcome the traditional problems of overestimating the probabilities of rare occurrences and underestimating the probabilities of unseen events, a range of smoothing algorithms have been proposed in the literature (Goodman, 2001). Most methods take a heuristic-frequentist approach combining n-gram probabilities for various values of n, using back-off schemes or interpolation.

Teh (2006) showed that MacKay and Peto's (1995) research on parametric Bayesian language models with a Dirichlet prior could be extended to give better results, but also that one of the best smoothing methods, interpolated Kneser-Ney (Kneser and Ney, 1995), can be derived as an approximation of the Hierarchical Pitman-Yor process language model (HPYLM).

The success of the Bayesian approach to language modelling is due to the use of statistical distributions such as the Dirichlet distribution, and distributions over distributions, such as the Dirichlet process and its two-parameter generalisation, the Pitman-Yor process. Both are widely studied in the statistics and probability theory communities. Interestingly, language modelling has acquired the status of a "fruit fly" problem in these communities, to benchmark the performance of statistical models. In this paper we approach language modelling from a computational linguistics point of view, and consider the statistical methods to be the tool with the future goal of improving language models for extrinsic tasks such as speech recognition.

We derive our model from Teh (2006), and propose an extension with skipgrams. A frequentist approach to language modelling with skipgrams is described by Pickhardt et al. (2014), who introduce an approach using skip-n-grams which are interpolated using modified Kneser-Ney smoothing. In this paper we show that a Bayesian skip-n-gram approach outperforms a frequentist skip-n-gram model.

2 Method

Traditionally, the most widely used pattern in language modelling is the n-gram, which represents

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 137–142,
Berlin, Germany, August 7-12, 2016.

a pattern of n contiguous words, of which we call the first $(n-1)$ words the history or context, and the nth word the focus word. The motivation for using n-grams can be traced back to the distributional hypothesis of Harris (Harris, 1954; Sahlgren, 2008). Although n-grams are small patterns without any explicit linguistic annotation, they are surprisingly effective in many tasks, such as language modelling in machine translation, automatic speech recognition, and information retrieval.

One of the main limitations of n-grams is their contiguity, because this limits the expressive power to relations between neighboring words. Many patterns in language span a range that is longer than the typical length of n; we call these relations long-distance relations. Other patterns may be within the range of n, but are still non-contiguous; they skip over positions. Both types of relations may be modelled with (syntactic) dependencies, and modelling these explicitly requires a method to derive a parser, e.g. a dependency parser, from linguistically annotated data.

To be able to model long-distance and other non-contiguous relations between words without resorting to explicitly computing syntactic dependencies, we use skipgrams. Skipgrams are a generalisation of n-grams. They consist of n tokens, but now each token may represent a skip of at least one word, where a skip can match any word. Let $\{m\}$ be a skip of length m, then *the* $\{1\}$ *house* can match "the big house", or "the yellow house", etc. We do not allow skips to be at the beginning or end of the skipgram, so for $n > 2$ skipgrams are a generalisation of n-grams (Goodman, 2001; Shazeer et al., 2015; Pickhardt et al., 2014).

Pitman-Yor Processes (PYP) belong to the family of non-parametric Bayesian models. Let W be a fixed and finite vocabulary of V words. For each word $w \in W$ let $G(w)$ be the probability of w, and $G = [G(w)]_{w \in W}$ be the vector of word probabilities. Since word frequencies generally follow a power-law distribution, we use a Pitman-Yor process, which is a distribution over partitions with power-law distributions. In the context of a language model this means that for a space $P(\mathbf{u})$, with $c(\mathbf{u}\cdot)$ elements (tokens), we want to partition $P(\mathbf{u})$ in V subsets such that the partition is a good approximation of the underlying data, in which $c(\mathbf{u}w)$ is the size of subset w of $P(\mathbf{u})$. We assume that the training data is an sample of the underlying data, and for this reason we seek to find an approximation, rather than using the partitions precisely as found in the training data.

Since we also assume that a power-law distribution on the words in the underlying data, we place a PYP prior on G:

$$G \sim \mathrm{PY}(d, \theta, G_0),$$

with discount parameter $0 \leq d < 1$, a strength parameter $\theta > -d$ and a mean vector $G_0 = [G_0(w)]_{w \in W}$. $G_0(w)$ is the a-priori probability of word w, which we set uniformly: $G_0(w) = 1/V$ for all $w \in W$. In general, there is no known analytic form for the density of $\mathrm{PY}(d, \theta, G_0)$ when the vocabulary is finite. However, we are interested in the distribution over word sequences induced by the PYP, which has a tractable form, and is sufficient for the purpose of language modelling.

Let G and G_0 be distributions over W, and $x_1, x_2, \ldots$ be a sequence of words drawn i.i.d. from G. The PYP is then described in terms of a generative procedure that takes $x_1, x_2, \ldots$ to produce a separate sequence of i.i.d. draws $y_1, y_2, \ldots$ from the mean distribution G_0 as follows. The first word x_1 is assigned the value of the first draw y_1 from G_0. Let t be the current number of draws from G_0, c_k the number of words assigned the value of draw y_k and $c_\cdot = \sum_{k=1}^{t} c_k$ the number of draws from G_0. For each subsequent word $x_{c_\cdot+1}$, we either assign it the value of a previous draw y_k, with probability $\frac{c_k - d}{\theta + c_\cdot}$, or assign it the value of a new draw from G_0 with probability $\frac{\theta + dt}{\theta + c_\cdot}$.

For an n-gram language model we use a hierarchical extension of the PYP. The hierarchical extension describes the probabilities over the current word given various contexts consisting of up to $n-1$ words. Given a context $\mathbf{u}$, let $G_{\mathbf{u}}(w)$ be the probability of the current word taking on value w. A PYP is used as the prior for $G_{\mathbf{u}} = [G_{\mathbf{u}}(w)]_{w \in W}$:

$$G_{\mathbf{u}} \sim \mathrm{PY}(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}),$$

where $\pi(\mathbf{u})$ is the suffix of $\mathbf{u}$ consisting of all but the first word, and $|\mathbf{u}|$ being the length of $\mathbf{u}$. The priors are recursively placed with parameters $\theta_{|\pi(\mathbf{u})|}$, $d_{|\pi(\mathbf{u})|}$ and mean vector $G_{\pi(\pi(\mathbf{u}))}$, until we get to $G_\emptyset$:

$$G_\emptyset \sim \mathrm{PY}(d_0, \theta_0, G_0),$$

with G_0 being the uniformly distributed global mean vector for the empty context $\emptyset$.

3 Backoff Strategies

In this paper we investigate three backoff strategies: ngram, limited, and full. ngram is the traditional n-gram backoff method as described by Teh (2006); limited and full are extensions that also incorporate skipgram probabilities. The full backoff strategy is similar to ngram in that it always backs off recursively to the word probabilities, while limited halts as soon as a probability is known for a pattern. The backoff strategies can be formalised as follows. For all strategies, we have that $p(w|\mathbf{u}) = G_0(w)$ if $\mathbf{u} = \emptyset$. For ngram, the other case is defined as:

$$p(w|\mathbf{u}) = \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|}t_{\mathbf{u}w\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}t_{\mathbf{u}\cdot\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} p(w|\pi(\mathbf{u}))$$

with $c_{\mathbf{u}w\cdot}$ being the number of $\mathbf{u}w$ tokens, and $c_{\mathbf{u}\cdot\cdot}$ the number of patterns starting with context $\mathbf{u}$. Similarly, $t_{\mathbf{u}wk}$ is 1 if draw the kth from $G_{\mathbf{u}}$ was w, 0 otherwise. $t_{\mathbf{u}w\cdot}$ then denotes if there is a pattern $\mathbf{u}w$, and $t_{\mathbf{u}\cdot\cdot}$ is the number of types following context $\mathbf{u}$.

Now let σ_n be the operator that adds a skip to a pattern $\mathbf{u}$ on the nth position if there is not already a skip. Then $\boldsymbol{\sigma}(\mathbf{u}) = [\sigma_n(\mathbf{u})]_{n=2}^{|\mathbf{u}|}$ is the set of patterns with one skip more than the number of skips currently in $\mathbf{u}$. The number of generated patterns is $\varsigma = |\boldsymbol{\sigma}(\mathbf{u})|$. We also introduce the indicator function S, which for the full backoff strategy always returns its argument: $S_{\mathbf{u}w}(y) = y$. The full backoff strategy is defined as follows, with $\mathbf{u}_x = \boldsymbol{\sigma}_x(\mathbf{u})$, and discount frequency $\delta_{\mathbf{u}} = 1$:

$$\begin{aligned} p(w|\mathbf{u}) = & \\ & \sum_{m=1}^{\varsigma} \left\{ \frac{1}{\varsigma+1} \left[\frac{c_{\mathbf{u}_m w\cdot} - \delta_{\mathbf{u}_m} d_{|\mathbf{u}_m|} t_{\mathbf{u}_m w\cdot}}{\delta_{\mathbf{u}_m}\theta_{|\mathbf{u}_m|} + c_{\mathbf{u}_m\cdot\cdot}} + \right.\right. \\ & \left.\left. S_{\mathbf{u}_m w} \left(\frac{\theta_{|\mathbf{u}_m|} + d_{|\mathbf{u}_m|} t_{\mathbf{u}_m\cdot\cdot}}{\delta_{\mathbf{u}_m}\theta_{|\mathbf{u}_m|} + c_{\mathbf{u}_m\cdot\cdot}} p(w|\pi(\mathbf{u}_m)) \right) \right] \right\} \\ & + \frac{1}{\varsigma+1} \left[\frac{c_{\mathbf{u}w\cdot} - \delta_{\mathbf{u}} d_{|\mathbf{u}|} t_{\mathbf{u}w\cdot}}{\delta_{\mathbf{u}}\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \right. \\ & \left. S_{\mathbf{u}w} \left(\frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot\cdot}}{\delta_{\mathbf{u}}\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} p(w|\pi(\mathbf{u})) \right) \right] \end{aligned}$$

The limited backoff strategy is an extension of the full backoff strategy that stops the recursion if a test pattern $\mathbf{u}w$ has already occurred in the training data. This means that the count is not zero, and hence at training time a probability has been assigned to that pattern. S is the indicator function which tells if a pattern has been seen during training: $S_{\mathbf{u}w}(\cdot) = 0$ if $\text{count}(\mathbf{u}w) > 0$, 1 otherwise; and $\delta_{\mathbf{u}} = V - \sum_{w \in W} S_{\mathbf{u}w}(\cdot)$. Setting $S_{\mathbf{u}w}(\cdot) = 0$ stops the recursion.

4 Data

In this section we give an overview of the data sets we use for the English and Flemish-Dutch experiments.

4.1 English Data

For the experiments on English we use four corpora: two large generic mixed-domain corpora and two smaller domain-specific corpora. We train on the largest of the two mixed-domain corpora, and test on all four corpora.

The first generic corpus is the Google 1 billion words shuffled web corpus of 769 million tokens (Chelba et al., 2013). For training we use sets 1 through 100, out of the 101 available training sets; for testing we use all available 50 test sets (8M tokens). The second generic corpus, used as test data, is a Wikipedia snapshot (368M tokens) of November 2013 as used and provided by Pickhardt et al. (2014). The first domain-specific corpus is from JRC-Acquis v3.0 (Steinberger et al., 2006), which contains legislative text of the European Union (8M tokens). The second domain-specific corpus consists of documents from the European Medicines Agency, EMEA (Tiedemann, 2009). We shuffled all sentences, and selected 20% of them as the test set (3M tokens).

Since the HPYLM uses a substantial amount of memory, even with histogram-based sampling, we cannot model the complete 1bw data set without thresholding the patterns in the model. We used a high occurrence threshold of 100 on the unigrams, yielding 99,553 types that occur above this threshold. We use all n-grams and skipgrams that occurred at least twice, consisting of the included unigrams as focus words, with UNKs occupying the positions of words not in the vocabulary. Note that because these settings are different from models competing on this benchmark, the results in this paper cannot be compared to those results.

4.2 Flemish-Dutch Data

For the experiments on Flemish-Dutch data, we use the Mediargus corpus as training material. It

contains 5 years of newspaper texts from 12 Flemish newspapers and magazines, totaling 1.3 billion words.

For testing we use the Flemish part of the Spoken Dutch Corpus (CGN) (Oostdijk, 2000) (3.2M words), divided over 15 components, ranging from spontaneous speech to books read aloud. CGN also contains two components which are news articles and news, which from a domain perspective are similar to the training data of Mediargus. We report on each component separately.

Similarly to the 1bw models, we used a threshold on the word types, such that we have a similar size of vocabulary (100k types), which we produced with a threshold of 250. We used the same occurrence threshold of 2 on the n- and skipgrams.

5 Experimental Setup

We train 4-gram language model on the two training corpora, the Google 1 billion word benchmark and the Mediargus corpus. We do not perform any preprocessing on the data except tokenisation. The models are trained with a HPYLM. We do not use sentence beginning and end markers. The results for the ngram backoff strategy are obtained by training without skipgrams; for limited and full we added skipgram features during training.

At the core of our experimental framework we use cpyp,[1] which is an existing library for non-parametric Bayesian modelling with PY priors with histogram-based sampling (Blunsom et al., 2009). This library has an example application to showcase its performance with n-gram based language modelling. Limitations of the library, such as not natively supporting skipgrams, and the lack of other functionality such as thresholding and discarding of certain patterns, led us to extend the library with Colibri Core,[2] a pattern modelling library. Colibri Core resolves the limitations, and together the libraries are a complete language model that handles skipgrams: cococpyp.[3]

Each model is run for 50 iterations (without an explicit burn-in phase), with hyperparameters $\theta = 1.0$ and $\gamma = 0.8$. The hyperparameters are resampled every 30 iterations with slice sampling (Walker, 2007). We test each model on different test sets, and we collect their intrinsic performance by means of perplexity. Words in the test set that were unseen in the training data are ignored in computing the perplexity on test data.

[1] https://github.com/redpony/cpyp

[2] http://proycon.github.io/colibri-core/

[3] https://github.com/naiaden/cococpyp

Test	ngram	limited	↓%	full	↓%
1bw	171	**141**	**6**	199	-16
jrc	1232	994	19	**728**	**41**
emea	1749	1304	25	**1069**	**39**
wp	724	635	12	**542**	**25**

Table 1: Results of the full and limited backoff systems, trained on 1bw, tested on 1bw (in-domain), and cross-domain sets jrc, emea, and wp. ↓% is the relative reduction in perplexity for the column to its left.

Comp.	ngram	limited	↓%	full	↓%
a	1280	1116	13	**828**	**35**
b	847	785	7	**639**	**24**
c	1501	1272	15	**946**	**37**
d	1535	1306	15	**975**	**36**
f	708	647	9	**572**	**19**
g	479	445	7	**440**	**8**
h	1016	916	10	**718**	**29**
i	1075	990	8	**783**	**27**
j	469	**434**	**7**	442	6
k	284	**253**	**11**	333	-17
l	726	639	12	**629**	**13**
m	578	538	7	**512**	**11**
n	895	794	11	**664**	**26**
o	1017	887	13	**833**	**18**

Table 2: Results of the full and limited backoff systems, trained on Mediargus, tested on CGN. Components range from spontaneous (a) to non-spontaneous (o), with components j (news reports) and k (news) being in-domain for the training corpus, and the other components being out-of-domain. ↓% is the relative reduction in perplexity for the column to its left.

6 Results

The results are reported in terms of perplexity, in Table 1 for English, and in Table 2 for Flemish-Dutch. We computed baseline perplexity scores with SRILM (Stolcke, 2002) for 1bw. We used an interpolated modified Kneser-Ney language model, with Good-Turing discounting to mimic our thresholding options. Although the models are not comparable, this is arguably the closest approximation in SRILM of our HPYLM. For 1bw the baseline is 147; for jrc, emea, and wp, 1391, 1430, and 1403 respectively. In some cases the

baseline is better compared to the ngram backoff strategy. With adding skipgrams we always outperform the baseline, especially on the out-of-domain test sets.

We find that with large data sets adding skipgrams lowers the perplexity, for both languages, in both within- and cross-domain experiments. For English, we observe absolute perplexity reductions up to 680 (a relative reduction of 39%) in a cross-domain setting, and absolute perplexity reductions of 10 (relative reduction of 6%) in a within-domain setting. For Flemish-Dutch we observe similar results with absolute reductions up to 560 (relative reduction of 36%) and 31 (relative reduction 11%), respectively.

If we consider the three backoff strategies individually, we can see the following effects on both English and Flemish-Dutch data. In a within-domain experiment limited backoff is the best strategy. In a cross-domain setting, the full backoff strategy yields the lowest perplexity and largest perplexity reductions. In the first case, stopping the backoff when there is a pattern probability for the word and its context yields a more certain probability than when the probability is diffused by more uncertain backoff probabilities.

Upon inspection of the model sizes, we observe that the skipgram model contains almost five times as many parameters as the n-gram model. This difference is explained by the addition of skipgrams of length 3 and 4, and the bigrams and unigrams derived from these skipgrams. Each 4-gram can be deconstructed into three skipgrams of length 4, and one of these skipgrams yields a skipgram of length 3. Tests with ngram backoff on skipgram models show that the performance is worse compared to ngram backoff in pure n-gram models because of the extra bigrams and unigrams (ngram ignores the skipgrams). Yet, the experimental results also indicate that with sufficient data, skipgram models outperform n-gram models. Because the difference in parameters is only noticeable in terms of memory, and it hardly impacts the run-time, this makes the skipgram model the favourable model.

7 Conclusions

In this paper we showed that by adding skipgrams, a straightforward but powerful generalisation of n-gram word patterns, we can reduce the perplexity of a Bayesian language model, especially in a cross-domain language modelling task. By changing the backoff strategy we can also improve on a within-domain task. We found this effect in two languages.

References

P Blunsom, T Cohn, S Goldwater, and M Johnson. 2009. A note on the implementation of hierarchical Dirichlet processes. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 337–340. Association for Computational Linguistics.

C Chelba, T Mikolov, M Schuster, Q Ge, T Brants, P Koehn, and T Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. Technical Report Google Tech Report 41880.

JT Goodman. 2001. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434.

ZS Harris. 1954. Distributional structure. *Word*.

R Kneser and H Ney. 1995. Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, May.

DJC MacKay and LCB Peto. 1995. A hierarchical Dirichlet language model. *Natural language engineering*, 1(3):289–308.

N Oostdijk. 2000. The spoken Dutch corpus. Overview and first evaluation. In *LREC*.

R Pickhardt, T Gottron, M Körner, PG Wagner, T Speicher, and S Staab. 2014. A generalized language model as the combination of skipped n-grams and modified Kneser-Ney smoothing. *arXiv preprint arXiv:1404.3377*.

M Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54.

N Shazeer, J Pelemans, and C Chelba. 2015. Sparse non-negative matrix language modeling for skipgrams. In *Proceedings of Interspeech*, pages 1428–1432.

R Steinberger, B Pouliquen, A Widiger, C Ignat, T Erjavec, D Tufis, and D Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation*.

A Stolcke. 2002. SRILM — an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, November.

YW Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992. Association for Computational Linguistics.

J Tiedemann. 2009. News from OPUS — A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume 5, pages 237–248.

SG Walker. 2007. Sampling the Dirichlet mixture model with slices. *Communications in Statistics — Simulation and Computation*, 36(1):45–54.