Hierarchical Bayesian Language Modelling for the Linguistically Informed

Jan A. Botha
Department of Computer Science
University of Oxford, UK
jan.botha@cs.ox.ac.uk

Abstract

In this work I address the challenge of augmenting n-gram language models according to prior linguistic intuitions. I argue that the family of hierarchical Pitman-Yor language models is an attractive vehicle through which to address the problem, and demonstrate the approach by proposing a model for German compounds. In an empirical evaluation, the model outperforms the Kneser-Ney model in terms of perplexity, and achieves preliminary improvements in English-German translation.

1 Introduction

The importance of effective language models in machine translation (MT) and automatic speech recognition (ASR) is widely recognised. n-gram models, in particular ones using Kneser-Ney (KN) smoothing, have become the standard workhorse for these tasks. These models are not ideal for languages that have relatively free word order and/or complex morphology. The ability to encode additional linguistic intuitions into models that already have certain attractive properties is an important piece of the puzzle of improving machine translation quality for those languages. But despite their widespread use, KN n-gram models are not easily extensible with additional model components that target particular linguistic phenomena.

I argue in this paper that the family of hierarchical Pitman-Yor language models (HPYLM) (Teh, 2006; Goldwater et al., 2006) are suitable for investigations into more linguistically-informed n-gram language models. Firstly, the flexibility to specify arbitrary back-off distributions makes it easy to incorporate multiple models into a larger n-gram model. Secondly, the Pitman-Yor process prior (Pitman and Yor, 1997) generates distributions that are well-suited to a variety of power-law behaviours, as is often observed in language. Catering for a variety of those is important since the frequency distributions of, say, suffixes, could be quite different from that of words. KN smoothing is less flexibility in this regard. And thirdly, the basic inference algorithms have been parallelised (Huang and Renals, 2009), which should in principle allow the approach to still scale to large data sizes.

As a test bed, I consider compounding in German, a common phenomenon that creates challenges for machine translation into German.

2 Background and Related Work

n-gram language models assign probabilities to word sequences. Their key approximation is that a word is assumed to be fully determined by $n-1$ words preceding it, which keeps the number of independent probabilities to estimate in a range that is computationally attractive. This basic model structure, largely devoid of syntactic insight, is surprisingly effective at biasing MT and ASR systems toward more fluent output, given a suitable choice of target language.

But the real challenge in constructing n-gram models, as in many other probabilistic settings, is how to do smoothing, since the vast majority of linguistically plausible n-grams will occur rarely or be absent altogether from a training corpus, which often renders empirical model estimates misleading. The general picture is that probability mass must be shifted away from some events and redistributed across others.

The method of Kneser and Ney (1995) and

Proceedings of the EACL 2012 Student Research Workshop, pages 64–73,
Avignon, France, 26 April 2012.

its later modified version (Chen and Goodman, 1998) generally perform best at this smoothing, and are based on the idea that the number of distinct contexts a word appears in is an important factor in determining the probability of that word. Part of this smoothing involves discounting the counts of n-grams in the training data; the modified version uses different levels of discounting depending on the frequency of the count. These methods were designed with surface word distributions, and are not necessarily suitable for smoothing distributions of other kinds of surface units.

Bilmes and Kirchhoff (2003) proposed a more general framework for n-gram language modelling. Their Factored Language Model (FLM) views a word as a vector of features, such that a particular feature value is generated conditional on some history of preceding feature values. This allowed the inclusion of n-gram models over sequences of elements like PoS tags and semantic classes. In tandem, they proposed more complicated back-off paths; for example, trigrams can back-off to two underlying bigram distributions, one dropping the left-most context word and the other the right-most. With the right combination of features and back-off structure they got good perplexity reductions, and obtained some improvements in translation quality by applying these ideas to the smoothing of the bilingual phrase table (Yang and Kirchhoff, 2006).

My approach has some similarity to the FLM: both decompose surface word forms into elements that are generated from unrelated conditional distributions. They differ predominantly along two dimensions: the types of decompositions and conditioning possible, and my use of a particular Bayesian prior for handling smoothing.

In addition to the HPYLM for n-gram language modelling (Teh, 2006), models based on the Pitman-Yor process prior have also been applied to good effect in word segmentation (Goldwater et al., 2006; Mochihashi et al., 2009) and speech recognition (Huang and Renals, 2007; Neubig et al., 2010). The Graphical Pitman-Yor process enables branching back-off paths, which I briefly revisit in §7, and have proved effective in language model domain-adaptation (Wood and Teh, 2009). Here, I extend this general line of inquiry by considering how one might incorporate linguistically informed sub-models into the HPYLM framework.

3 Compound Nouns

I focus on compound nouns in this work for two reasons: Firstly, compounding is in general a very productive process, and in some languages (including German, Swedish and Dutch) they are written as single orthographic units. This increases data sparsity and creates significant challenges for NLP systems that use whitespace to identify their elementary modelling units. A proper account of compounds in terms of their component words therefore holds the potential of improving the performance of such systems.

Secondly, there is a clear linguistic intuition to exploit: the morphosyntactic properties of these compounds are often fully determined by the head component within the compound. For example, in "Geburtstagskind" (birthday kid), it is "Kind" that establishes this compound noun as singular neuter, which determine how it would need to agree with verbs, articles and adjectives. In the next section, I propose a model in the suggested framework that encodes this intuition.

The basic structure of German compounds comprises a *head component*, preceded by one or more *modifier components*, with optional *linker elements* between consecutive components (Goldsmith and Reutter, 1998).

Examples

- The basic form is just the concatenation of two nouns

  Auto + Unfall = Autounfall (car crash)
- Linker elements are sometimes added between components

  Küche + Tisch = Küchentisch (kitchen table)
- Components can undergo stemming during composition

  Schule + Hof = Schulhof (schoolyard)
- The process is potentially recursive

  (Geburt + Tag) + Kind = Geburtstag + Kind

  = Geburtstagskind (birthday kid)

The process is not limited to using nouns as components, for example, the numeral in Zwei-Euro-Münze (two Euro coin) or the verb "fahren" (to drive) in Fahrzeug (vehicle). I will treat all these cases the same.

3.1 Fluency amid sparsity

Consider the following example from the training corpus used in the subsequent evaluations:

de: Die *Neu*infektionen übersteigen weiterhin die *Behandlungs*bemühungen.

en: *New* infections continue to outpace *treatment* efforts.

The corpus contains numerous other compounds ending in "infektionen" (16) or "bemühungen" (117). A standard word-based n-gram model discriminates among those alternatives using as many independent parameters.

However, we could gauge the approximate syntactic fluency of the sentence almost as well if we ignore the compound modifiers. Collapsing all the variants in this way reduces sparsity and yields better n-gram probability estimates.

To account for the compound modifiers, a simple approach is to use a reverse n-gram language model over compound components, without conditioning on the sentential context. Such a model essentially answers the question, "Given that the word ends in 'infektionen', what modifier(s), if any, are likely to precede it?" The vast majority of nouns will never occur in that position, meaning that the conditional distributions will be sharply peaked.

Figure 1: Intuition for the proposed generative process of a compound word: The context generates the head component, which generates a modifier component, which in turn generates another modifier. (Translation: "with the cable car")

3.2 Related Work on Compounds

In machine translation and speech recognition, one approach has been to split compounds as a preprocessing step and merge them back together during postprocessing, while using otherwise unmodified NLP systems. Frequency-based methods have been used for determining how aggressively to split (Koehn and Knight, 2003), since the maximal, linguistically correct segmentation is not necessarily optimal for translation. This gave rise to slight improvements in machine translation evaluations (Koehn et al., 2008), with fine-tuning explored in (Stymne, 2009). Similar ideas have also been employed for speech recognition (Berton et al., 1996) and predictive-text input (Baroni and Matiasek, 2002), where single-token compounds also pose challenges.

4 Model Description

4.1 HPYLM

Formally speaking, an n-gram model is an $(n-1)$-th order Markov model that approximates the joint probability of a sequence of words $\mathbf{w}$ as

$$P(\mathbf{w}) \approx \prod_{i=1}^{|\mathbf{w}|} P(w_i|w_{i-n+1},\ldots,w_{i-1}),$$

for which I will occasionally abbreviate a context $[w_i,\ldots,w_j]$ as $\mathbf{u}$. In the HPYLM, the conditional distributions $P(w|\mathbf{u})$ are smoothed by placing Pitman-Yor process priors (PYP) over them. The PYP is defined through its base distribution, and a *strength* (θ) and *discount* (d) hyperparameter that control its deviation away from its mean (which equals the base distribution).

Let $G_{[u,v]}$ be the PYP-distributed trigram distribution $P(w|u,v)$. The hierarchy arises by using as base distribution for the prior of $G_{[u,v]}$ another PYP-distributed $G_{[v]}$, i.e. the distribution $P(w|v)$. The recursion bottoms out at the unigram distribution $G_\emptyset$, which is drawn from a PYP with base distribution equal to the uniform distribution over the vocabulary $\mathcal{W}$. The hyperparameters are tied across all priors with the same context length $|\mathbf{u}|$, and estimated during training.

$$\begin{aligned}
G_0 &= \text{Uniform}(|\mathcal{W}|)\\
G_\emptyset &\sim PY(d_0,\theta_0,G_0)\\
&\vdots\\
G_{\pi(\mathbf{u})} &\sim PY(d_{|\pi(\mathbf{u})|},\theta_{|\pi(\mathbf{u})|},G_{\pi(\pi(\mathbf{u}))})\\
G_{\mathbf{u}} &\sim PY(d_{|\mathbf{u}|},\theta_{|\mathbf{u}|},G_{\pi(\mathbf{u})})\\
w &\sim G_{\mathbf{u}},
\end{aligned}$$

where $\pi(\mathbf{u})$ truncates the context $\mathbf{u}$ by dropping the left-most word in it.

4.2 HPYLM+c

Define a compound word $\tilde{w}$ as a sequence of components $[c_1,\ldots,c_k]$, plus a sentinel symbol \$ marking either the left or the right boundary of the word, depending on the direction of the model. To maintain generality over this choice of direction,

let Λ be an index set over the positions, such that c_{Λ_1} always designates the head component.

Following the motivation in §3.1, I set up the model to generate the head component c_{Λ_1} conditioned on the word context $\mathbf{u}$, while the remaining components $\tilde{w} \setminus c_{\Lambda_1}$ are generated by some model F, independently of $\mathbf{u}$.

To encode this, I modify the HPYLM in two ways: 1) Replace the support with the reduced vocabulary $\mathcal{M}$, the set of unique components c obtained when segmenting the items in $\mathcal{W}$. 2) Add an additional level of conditional distributions $H_{\mathbf{u}}$ (with $|\mathbf{u}| = n-1$) where items from $\mathcal{M}$ combine to form the observed surface words:

$$
\begin{aligned}
G_{\mathbf{u}} &\ \ldots \text{ (as before, except } G_0 = \text{Uniform}(|\mathcal{M}|)) \\
H_{\mathbf{u}} &\sim PY(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\mathbf{u}} \times F) \\
\tilde{w} &\sim H_{\mathbf{u}}
\end{aligned}
$$

So the base distribution for the prior of the word n-gram distribution $H_{\mathbf{u}}$ is the product of a distribution $G_{\mathbf{u}}$ over compound heads, given the same context $\mathbf{u}$, and another (n′-gram) language model F over compound modifiers, conditioned on the head component.

Choosing F to be a bigram model (n′=2) yields the following procedure for generating a word:

$$
\begin{aligned}
c_{\Lambda_1} &\sim G_{\mathbf{u}} \\
\text{for } i &= 2 \text{ to } k \\
c_{\Lambda_i} &\sim F(\cdot|c_{\Lambda_{i-1}})
\end{aligned}
$$

The linguistically motivated choice for conditioning in F is $\Lambda^{\text{ling}} = [k, k-1, \ldots, 1]$ such that c_{Λ_1} is the true head component; \$ is drawn from $F(\cdot|c_1)$ and marks the left word boundary.

In order to see if the correct linguistic intuition has any bearing on the model's extrinsic performance, we will also consider the reverse, supposing that the left-most component were actually more important in this task, and letting the remaining components be generated left-to-right. This is expressed by $\Lambda^{\text{inv}} = [1, \ldots, k]$, where \$ this time marks the right word boundary and is drawn from $F(\cdot|c_k)$.

To test whether Kneser-Ney smoothing is indeed sometimes less appropriate, as conjectured earlier, I will also compare the case where $F = F_{KN}$, a KN-smoothed model, with the case where $F = F_{HPYLM}$, another HPYLM.

Linker Elements In the preceding definition of compound segmentation, the linker elements do not form part of the vocabulary $\mathcal{M}$. Regarding linker elements as components in their own right would sacrifice important contextual information and disrupt the conditionals $F(\cdot|c_{\Lambda_{i-1}})$. That is, given Küche·n·tisch, we want $P(\text{Küche}|\text{Tisch})$ in the model, but not $P(\text{Küche}|\text{n})$.

But linker elements need to be accounted for somehow to have a well-defined generative model. I follow the pragmatic option of merging any linkers onto the adjacent component – for Λ^{ling} merging happens onto the preceding component, while for Λ^{inv} it is onto the succeeding one. This keeps the 'head' component c_{Λ_1} in tact.

More involved strategies could be considered, and it is worth noting that for German the presence and identity of linker elements between c_i and c_{i+1} are in fact governed by the preceding component c_i (Goldsmith and Reutter, 1998).

5 Training

For ease of exposition I describe inference with reference to the trigram HPYLM+c model with a bigram HPYLM for F, but the general case should be clear.

The model is specified by the latent variables $(G_{[\emptyset]}, G_{[v]}, G_{[u,v]}, H_{[u,v]}, F_{\emptyset}, F_c)$, where $u, v \in \mathcal{W}$, $c \in \mathcal{M}$, and hyperparameters $\Omega = \{d_i, \theta_i\} \cup \{d'_j, \theta'_j\} \cup \{d''_2, \theta''_2\}$, where $i = 0, 1, 2$, $j = 0, 1$, single primes designate the hyperparameters in F_{HPYLM} and double primes those of $H_{[u,v]}$. We can construct a collapsed Gibbs sampler by marginalising out these latent variables, giving rise to a variant of the hierarchical Chinese Restaurant Process in which it is straightforward to do inference.

Chinese Restaurant Process A direct representation of a random variable G drawn from a PYP can be obtained from the so-called stick-breaking construction. But the more indirect representation by means of the Chinese Restaurant Process (CRP) (Pitman, 2002) is more suitable here since it relates to distributions over items drawn from such a G. This fits the current setting, where words w are being drawn from a PYP-distributed G.

Imagine that a corpus is created in two phases: Firstly, a sequence of blank tokens x_i is instantiated, and in a second phase lexical identities w_i are assigned to these tokens, giving rise to the

observed corpus. In the CRP metaphor , the sequence of tokens x_i are equated with a sequence of customers that enter a restaurant one-by-one to be seated at one of an infinite number of tables. When a customer sits at an unoccupied table k, they order a dish ϕ_k for the table, but customers joining an occupied table have to dine on the dish already served there. The dish ϕ_i that each customer eats is equated to the lexical identity w_i of the corresponding token, and the way in which tables and dishes are chosen give rise to the characteristic properties of the CRP:

More formally, let $x_1, x_2, \ldots$ be draws from G, while t is the number of occupied tables, c the number of customers in the restaurant, and c_k the number of customers at the k-th table. Conditioned on preceding customers $x_1, \ldots, x_{i-1}$ and their arrangement, the i-th customer sits at table $k = k'$ according to the following probabilities:

$$\Pr(k'|\ldots) \propto \begin{cases} c_{k'} - d & \text{occupied table } k' \\ \theta + dt & \text{unoccupied table } t+1 \end{cases}$$

Ordering a dish for a new table corresponds to drawing a value ϕ_k from the base distribution G_0, and it is perfectly acceptable to serve the same kind of dish at multiple tables.

Some characteristic behaviour of the CRP can be observed easily from this description: 1) As more customers join a table, that table becomes a more likely choice for future customers too. 2) Regardless of how many customers there are, there is always a non-zero probability of joining an unoccupied table, and this probability also depends on the number of total tables.

The dish draws can be seen as backing off to the underlying base distribution G_0, an important consideration in the context of the hierarchical variant of the process explained shortly. Note that the strength and discount parameters control the extent to which new dishes are drawn, and thus the extent of reliance on the base distribution.

The predictive probability of a word w given a seating arrangement is given by

$$\Pr(w|\ldots) \propto c_w - dt_w + (\theta + dt)G_0(w)$$

In smoothing terminology, the first term can be interpreted as applying a discount of dt_w to the observed count c_w of w; the amount of discount therefore depends on the prevalence of the word (via t_w). This is one significant way in which the PYP/CRP gives more nuanced smoothing than modified Kneser-Ney, which only uses four different discount levels (Chen and Goodman, 1998). Similarly, if the seating dynamics are constrained such that each dish is only served once ($t_w = 1$ for any w), a single discount level is affected, establishing direct correspondence to original interpolated Kneser-Ney smoothing (Teh, 2006).

Hierarchical CRP When the prior of $G_{\mathbf{u}}$ has a base distribution $G_{\pi(\mathbf{u})}$ that is itself PYP-distributed, as in the HPYLM, the restaurant metaphor changes slightly. In general, each node in the hierarchy has an associated restaurant. Whenever a new table is opened in some restaurant R, another customer is plucked out of thin air and sent to join the parent restaurant pa(R). This induces a consistency constraint over the hierarchy: the number of tables t_w in restaurant R must equal the number of customers c_w in its parent pa(R).

In the proposed HPYLM+c model using F_{HPYLM}, there is a further constraint of a similar nature: When a new table is opened and serves dish $\phi = \tilde{w}$ in the trigram restaurant for $H_{[u,v]}$, a customer c_{Λ_1} is sent to the corresponding bigram restaurant for $G_{[u,v]}$, *and* customers $c_{\Lambda_{2:k}}$,\$ are sent to the restaurants for $F_{c'}$, for contexts $c' = c_{\Lambda_{1:k-1}}$. This latter requirement is novel here compared to the hierarchical CRP used to realise the original HPYLM.

Sampling Although the CRP allows us to replace the priors with seating arrangements S, those seating arrangements are simply latent variables that need to be integrated out to get a true predictive probability of a word:

$$p(w|\mathcal{D}) = \int_{S,\Omega} p(w|S,\Omega)p(S,\Omega|\mathcal{D}),$$

where $\mathcal{D}$ is the training data and, as before, Ω are the parameters. This integral can be approximated by averaging over m posterior samples (S,Ω) generated using Markov chain Monte Carlo methods. The simple form of the conditionals in the CRP allows us to do a Gibbs update whereby the table index k of a customer is resampled conditioned on all the other variables. Sampling a new seating arrangement S for the trigram HPYLM+c thus corresponds to visiting each customer in the restaurants for $H_{[u,v]}$, removing them while cascading as necessary to observe the consistency

across the hierarchy, and seating them anew at some table k'.

In the absence of any strong intuitions about appropriate values for the hyperparameters, I place vague priors over them and use slice sampling[1] (Neal, 2003) to update their values during generation of the posterior samples:

$$d \sim \text{Beta}(1,1) \qquad \theta \sim \text{Gamma}(1,1)$$

Lastly, I make the further approximation of $m = 1$, i.e. predictive probabilities are informed by a single posterior sample (S, Ω).

6 Experiments

The aim of the experiments reported here is to test whether the richer account of compounds in the proposed language models has positive effects on the predictability of unseen text and the generation of better translations.

6.1 Methods

Data and Tools Standard data preprocessing steps included normalising punctuation, tokenising and lowercasing all words. All data sets are from the WMT11 shared-task.[2]. The full English-German bitext was filtered to exclude sentences longer than 50, resulting in 1.7 million parallel sentences; word alignments were inferred from this using the Berkeley Aligner (Liang et al., 2006) and used as basis from which to extract a Hiero-style synchronous CFG (Chiang, 2007).

The weights of the log-linear translation models were tuned towards the BLEU metric on development data using `cdec`'s (Dyer et al., 2010) implementation of MERT (Och, 2003). For this, the set `news-test2008` (2051 sentences) was used, while final case-insensitive BLEU scores are measured on the official test set `newstest2011` (3003 sentences).

All language models were trained on the target side of the preprocessed bitext containing 38 million tokens, and tested on all the German development data (i.e. `news-test2008,9,10`).

Compound segmentation To construct a segmentation dictionary, I used the 1-best segmentations from a supervised MaxEnt compound splitter (Dyer, 2009) run on all token types in bitext. In addition, word-internal hyphens were also taken as segmentation points. Finally, linker elements were merged onto components as discussed in §4.2. Any token that is split into more than one part by this procedure is regarded as a compound. The effect of the individual steps is summarised in Table 1.

	# Types	Example
None	350998	Geburtstagskind
pre-merge	201328	Geburtstag·kind
merge, Λ^{ling}	150980	Geburtstags·kind
merge, Λ^{inv}	162722	Geburtstag·skind

Table 1: Effect of segmentation on vocabulary size.

Metrics For intrinsic evaluation of language models, perplexity is a common metric. Given a trained model q, the perplexity over the words τ in unseen test set T is $\exp\left(-\frac{1}{|T|}\sum_\tau \ln(q(\tau))\right)$.

One convenience of this per-word perplexity is that it can be compared consistently across different test sets regardless of their lengths; its neat interpretation is another: a model that achieves a perplexity of η on a test set is on average η-ways confused about each word. Less confusion and therefore lower test set perplexity is indicative of a better model. This allows different models to be compared relative to the same test set.

The exponent above can be regarded as an approximation of the cross-entropy between the model q and a hypothetical model p from which both the training and test set were putatively generated. It is sometimes convenient to use this as an alternative measure.

But a language model only really becomes useful when it allows some extrinsic task to be executed better. When that extrinsic task is machine translation, the translation quality can be assessed to see if one language model aids it more than another. The obligatory metric for evaluating machine translation quality is BLEU (Papineni et al., 2001), a precision based metric that measures how close the machine output is to a known correct translation (the reference sentences in the test set). Higher precision means the translation system is getting more phrases right.

Better language model perplexities sometimes lead to improvements in translation quality, but it is not guaranteed. Moreover, even when real translation improvements are obtained, they are

[1] Mark Johnson's implementation, `http://www.cog.brown.edu/~mj/Software.htm`

[2] `http://www.statmt.org/wmt11/`

	PPL	c-Cross-ent.
mKN	441.32	**0.1981**
HPYLM	429.17	0.1994
F_{KN} Λ^{ling}	432.95	0.2028
F_{KN} Λ^{inv}	446.84	0.2125
F_{HPYLM} Λ^{ling}	**421.63**	0.1987
F_{HPYLM} Λ^{inv}	435.79	0.2079

Table 2: Monolingual evaluation results. The second column shows perplexity measured all WMT11 German development data (7065 sentences). At the word level, all are trigram models, while F are bigram models using the specified segmentation scheme. The third column has test cross-entropies measured only on the 6099 compounds in the test set (given their contexts).

not guaranteed to be noticeable in the BLEU score, especially when targeting an arguably narrow phenomenon like compounding.

	BLEU
mKN	13.11
HPYLM	13.20
F_{HPYLM}, Λ^{ling}	13.24
F_{HPYLM}, Λ^{inv}	**13.32**

Table 3: Translation results, BLEU (1-ref), 3003 test sentences. Trigram language models, no count pruning, no "unknown word" token.

	P / R / F
mKN	22.0 / 17.3 / 19.4
HPYLM	21.0 / 17.8 / 19.3
F_{HPYLM}, Λ^{ling}	23.6 / 17.3 / **19.9**
F_{HPYLM}, Λ^{inv}	24.1 / 16.5 / 19.6

Table 4: Precision, Recall and F-score of compound translations, relative to reference set (72661 tokens, of which 2649 are compounds).

6.2 Main Results

For the monolingual evaluation, I used an interpolated, modified Kneser-Ney model (mKN) and an HPYLM as baselines. It has been shown for other languages that HPYLM tends to outperform mKN (Okita and Way, 2010), but I am not aware of this result being demonstrated on German before, as I do in Table 2.

The main model of interest is HPYLM+c using the Λ^{ling} segmentation and a model F_{HPYLM} over modifiers; this model achieves the lowest perplexity, 4.4% lower than the mKN baseline.

Next, note that using F_{KN} to handle the modifiers does worse than F_{HPYLM}, confirming our expectation that KN is less appropriate for that task, although it still does better than the original mKN baseline.

The models that use the linguistically implausible segmentation scheme Λ^{inv} both fare worse than their counterparts that use the sensible scheme, but of all tested models only F_{KN} & Λ^{inv} fails to beat the mKN baseline. This suggests that in some sense having *any* account whatsoever of compound formation tends to have a beneficial effect on this test set – the richer statistics due to a smaller vocabulary could be sufficient to explain this – but to get the most out of it one needs the superior smoothing over modifiers (provided by F_{HPYLM}) and adherence to linguistic intuition (via Λ^{ling}).

As for the translation experiments, the relative qualitative performance of the two baseline language models carries over to the BLEU score (HPYLM does 0.09 points better than KN), and is further improved upon slightly by using two variants of HPYLM+c (Table 3).

6.3 Analysis

To get a better idea of how the extended models employ the increased expressiveness, I calculated the cross-entropy over only the compound words in the monolingual test set (second column of Table 2). Among the HPYLM+c variants, we see that their performance on compounds only is consistent with their performance (relative to each other) on the whole corpus. This implies that the differences in whole-corpus perplexities are at least in part due to their different levels of adeptness at handling compounds, as opposed to some fluke event.

It is, however, somewhat surprising to observe that HPYLM+c do not achieve a lower compound cross-entropy than the mKN baseline, as it suggests that HPYLM+c's perplexity reductions compared to mKN arise in part from something other than compound handling, which is their whole point.

This discrepancy could be related to the fairness of this direct comparison of models that ul-

timately model different sets of things: According to the generative process of HPYLM+c (§4), there is no limit on the number of components in a compound: in theory, an arbitrary number of components $c \in \mathcal{M}$ can combine to form a word. HPYLM+c is thus defined over a countably infinite set of words, thereby reserving some probability mass for items that will never be realised in any corpus, whereas the baseline models are defined only over the finite set $\mathcal{W}$. These direct comparisons are thus lightly skewed in favour of the baselines. This bolsters confidence in the perplexity reductions presented in the previous section, but the skew may afflict compounds more starkly, leading to the slight discrepancy observed in the compound cross-entropies. What matters more is the performance among the HPYLM+c variants, since they are directly comparable.

To home in still further on the compound modelling, I selected those compounds for which HPYLM+c ($F_{HPYLM}, \Lambda^{\text{ling}}$) does best/worst in terms of the probabilities assigned, compared to the mKN baseline (see Table 5). One pattern that emerges is that the "top" compounds mostly consist of components that are likely to be quite common, and that this improves estimates both for n-grams that are very rare (the singleton "senkungen der treibhausgasemmissionen" = *decreases in green house gas emissions*) or relatively common (158, "der hauptstadt" = *of the capital*).

n-gram	Δ	C
gesichts·punkten	0.064	335
700 milliarden us-·dollar	0.021	2
s. der treibhausgas·emissionen	0.018	1
r. der treibhausgas·emissionen	0.011	3
ministerium für land·wirtschaft	0.009	11
bildungs·niveaus	0.009	14
newt ging·rich*	-0.257	2
nouri al-·maliki*	-0.257	3
klerikers moqtada al-·sadr*	-0.258	1
nuri al-·maliki*	-0.337	3
sankt peters·burg*	-0.413	35
nächtlichem flug·lärm	-0.454	2

Table 5: Compound n-grams in the test set for which the absolute difference $\Delta = P_{\text{HPYLM+c}} - P_{\text{mKN}}$ is greatest. **C** is n-gram count in the training data. Asterisks denote words that are not compounds, linguistically speaking. Abbrevs: r. = reduktionen, s.= senkungen

On the other hand, the "bottom" compounds are mostly ones whose components will be uncommon; in fact, many of them are not truly compounds but artefacts of the somewhat greedy segmentation procedure I used. Alternative procedures will be tested in future work.

Since the BLEU scores do not reveal much about the new language models' effect on compound translation, I also calculated compound-specific accuracies, using precision, recall and F-score (Table 4). Here, the precision for a single sentence would be 100% if all the compounds in the output sentence occur in the reference translation. Compared to the baselines, the compound precision goes up noticeably under the HPYLM+c models used in translation, without sacrificing on recall. This suggests that these models are helping to weed out incorrectly hypothesised compounds.

6.4 Caveats

All results are based on single runs and are therefore not entirely robust. In particular, MERT tuning of the translation model is known to introduce significant variance in translation performance across different runs, and the small differences in BLEU scores reported in Table 3 are very likely to lie in that region.

Markov chain convergence also needs further attention. In absence of complex latent structure (for the dishes), the chain should mix fairly quickly, and as attested by Figure 2 it 'converges' with respect to the test metric after about 20 samples, although the log posterior (not shown) had not converged after 40. The use of a single posterior sample could also be having a negative effect on results.

7 Future Directions

The first goal will be to get more robust experimental results, and to scale up to 4-gram models estimated on all the available monolingual training data. If good performance can be demonstrated under those conditions, this general approach could pass as a viable alternative to the current Kneser-Ney dominated state-of-the art setup in MT.

Much of the power of the HPYLM+c model has not been exploited in this evaluation, in particular its ability to score unseen compounds consisting of known components. This feature was

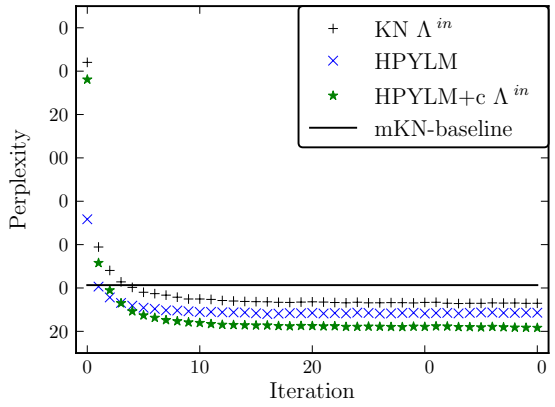

Figure 2: Convergence of test set perplexities.

not active in these evaluations, mostly due to the current phase of implementation. A second area of focus is thus to modify the decoder to generate such unseen compounds in translation hypotheses. Given the current low compound recall rates, this could greatly benefit translation quality. An informal analysis of the reference translations in the bilingual test set showed that 991 of the 1406 out-of-vocabulary compounds (out of 2692 OOVs in total) fall into this category of unseen-but-recognisable compounds.

Ultimately the idea is to apply this modelling approach to other linguistic phenomena as well. In particular, the objective is to model instances of concatenative morphology beyond compounding, with the aim of improving translation into morphologically rich languages. Complex agreement patterns could be captured by conditioning functional morphemes in the target word on morphemes in the n-gram context, or by stemming context words during back-off. Such additional back-off paths can be readily encoded in the Graphical Pitman-Yor process (Wood and Teh, 2009).

These more complex models may require longer to train. To this end, I intend to use the single table per dish approximation (§5) to reduce training to a single deterministic pass through the data, conjecturing that this will have little effect on extrinsic performance.

8 Summary

I have argued for further explorations into the use of a family of hierarchical Bayesian models for targeting linguistic phenomena that may not be captured well by standard n-gram language models. To ground this investigation, I focused on German compounds and showed how these models are an appropriate vehicle for encoding prior linguistic intuitions about such compounds. The proposed generative model beats the popular modified Kneser-Ney model in monolingual evaluations, and preliminarily achieves small improvements in translation from English into German. In this translation task, single-token German compounds traditionally pose challenges to translation systems, and preliminary results show a small increase in the F-score accuracy of compounds in the translation output. Finally, I have outlined the intended steps for expanding this line of inquiry into other related linguistic phenomena and for adapting a translation system to get optimal value out of such improved language models.

Acknowledgements

Thanks goes to my supervisor, Phil Blunsom, for continued support and advice; to Chris Dyer for suggesting the focus on German compounds and supplying a freshly trained compound splitter; to the Rhodes Trust for financial support; and to the anonymous reviewers for their helpful feedback.

References

Marco Baroni and Johannes Matiasek. 2002. Predicting the components of German nominal compounds. In *ECAI*, pages 470–474.

Andre Berton, Pablo Fetter, and Peter Regel-Brietzmann. 1996. Compound Words in Large-Vocabulary German Speech Recognition Systems. In *Proceedings of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 2, pages 1165–1168. IEEE.

Jeff A Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel back-off. In *Proceedings of NAACL-HLT (short papers)*, pages 4–6, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stanley F Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical report.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228, June.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A Decoder, Alignment, and Learning framework for finite-state and context-free translation models. In *Proceedings of the Association*

for Computational Linguistics (Demonstration session), pages 7–12, Uppsala, Sweden. Association for Computational Linguistics.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *Proceedings of NAACL*, pages 406–414. Association for Computational Linguistics.

John Goldsmith and Tom Reutter. 1998. Automatic Collection and Analysis of German Compounds. In F. Busa F. et al., editor, *The Computational Treatment of Nominals*, pages 61–69. Universite de Montreal, Canada.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Interpolating Between Types and Tokens by Estimating Power-Law Generators. In *Advances in Neural Information Processing Systems, Volume 18*.

Songfang Huang and Steve Renals. 2007. Hierarchical Pitman-Yor Language Models For ASR in Meetings. *IEEE ASRU*, pages 124–129.

Songfang Huang and Steve Renals. 2009. A parallel training algorithm for hierarchical Pitman-Yor process language models. In *Proceedings of Interspeech*, volume 9, pages 2695–2698.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modelling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and SIgnal Processing*, pages 181–184.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of EACL*, pages 187–193. Association for Computational Linguistics.

Philipp Koehn, Abhishek Arun, and Hieu Hoang. 2008. Towards better Machine Translation Quality for the German – English Language Pairs. In *Third Workshop on Statistical Machine Translation*, number June, pages 139–142. Association for Computational Linguistics.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June. Association for Computational Linguistics.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - ACL-IJCNLP '09*, pages 100–108, Suntec, Singapore. Association for Computational Linguistics.

Radford M Neal. 2003. Slice Sampling. *The Annals of Statistics*, 31(3):705–741.

Graham Neubig, Masato Mimura, Shinsuke Mori, and Tatsuya Kawahara. 2010. Learning a Language Model from Continuous Speech. In *Interspeech*, pages 1053–1056, Chiba, Japan.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*, pages 160–167.

Tsuyoshi Okita and Andy Way. 2010. Hierarchical Pitman-Yor Language Model for Machine Translation. *Proceedings of the International Conference on Asian Language Processing*, pages 245–248.

Kishore Papineni, Salim Roukos, Todd Ward, Weijing Zhu, Thomas J Watson, and Yorktown Heights. 2001. Bleu: A Method for Automatic Evaluation of Machine Translation. Technical report, IBM.

J Pitman and M. Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25:855–900.

J. Pitman. 2002. Combinatorial stochastic processes. Technical report, Department of Statistics, University of California at Berkeley.

Sara Stymne. 2009. A comparison of merging strategies for translation of German compounds. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 61–69.

Yee Whye 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 ACL*, pages 985–992. Association for Computational Linguistics.

Frank Wood and Yee Whye Teh. 2009. A Hierarchical Nonparametric Bayesian Approach to Statistical Language Model Domain Adaptation. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 607–614, Clearwater Beach, Florida, USA.

Mei Yang and Katrin Kirchhoff. 2006. Phrase-based Backoff Models for Machine Translation of Highly Inflected Languages. In *Proceedings of the EACL*, pages 41–48.