Bayesian Language Modelling of German Compounds

Jan A. Botha[1] *Chris Dyer*[2] *Phil Blunsom*[1]
(1) University of Oxford
(2) Carnegie Mellon University
{jan.botha,phil.blunsom}@cs.ox.ac.uk, cdyer@cs.cmu.edu

ABSTRACT
In this work we address the challenge of augmenting *n*-gram language models according to prior linguistic intuitions. We 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 our empirical evaluation the model outperforms a modified Kneser-Ney *n*-gram model in test set perplexity. When used as part of a translation system, the proposed language model matches the baseline BLEU score for English→German while improving the precision with which compounds are output. We find that an approximate inference technique inspired by the Bayesian interpretation of Kneser-Ney smoothing (Teh, 2006) offers a way to drastically reduce model training time with negligible impact on translation quality.

TITLE AND ABSTRACT IN AFRIKAANS

Bayes-modellering van saamgestelde woorde in Duits

Hierdie werk neem uitdagings rondom die uitbreiding van *n*-gramtaalmodelle volgens voorafgaande linguistieke intuïsie onder die loep. Ons voer aan dat die familie van hiërargiese Pitman-Yor taalmodelle 'n wenslike stuk gereedskap is om hierdie probleem mee aan te pak en formuleer 'n model van Duitse saamgestelde woorde om die benadering te demonstreer. Met behulp van 'n empiriese evaluering bevind ons dat die model in terme van toetsdataperpleksiteit beter vaar as die aangepaste Kneser-Ney *n*-grammodel. As onderdeel van 'n Engels→Duits-vertalingstelsel behaal die model in terme van die BLEU-metriek dieselfde vertaalafvoerkwaliteit as die kontrole stelsel en genereer saamgestelde woorde teen 'n hoër presisie. Verder stel ons vas dat 'n benaderde inferensietegniek, geïnspireer deur die Bayes-interpretasie van Kneser-Ney-gladstryking (Teh, 2006), gebruik kan word om die modelberamingtyd drasties te verminder sonder wesenlike impak op die vertaalafvoerkwaliteit.

KEYWORDS: language model; Bayesian methods; machine translation; compounding; ngram model; approximate inference.

KEYWORDS IN L_2: taalmodel; Bayes-metodes; masjienvertaling; samestellings; ngrammodel; benaderde inferensie.

Proceedings of COLING 2012: Technical Papers, pages 341–356,
COLING 2012, Mumbai, December 2012.

1 Introduction

Statistical language modelling addresses the problem of assigning probabilities to sentences in natural languages. In an effective model, these probabilities function as statistical proxies for sentences' syntactic well-formedness and semantic plausibility. As such, language models (LMs) play a crucial role in machine translation (MT) and automatic speech recognition (ASR) systems, which need to distinguish well-formed output sentences from ill-formed ones.

To tackle the problem of assigning reasonable probabilities to an infinite space of possible sentences, two assumptions are commonly made: first, the closed vocabulary assumption, which states that sentences are sequences of words from a *finite* vocabulary $\mathscr{W}$, and second, a Markov assumption is made, which states that the probability of each word in a sentence is conditionally independent of all others, given the previous $n-1$ words of context. Relying on these assumptions, language modelling becomes the problem of estimating the conditional probabilities of $|\mathscr{W}|$ words in $|\mathscr{W}|^{n-1}$ contexts.

There are two problems with this approach that we address in this paper.[1] First, the closed vocabulary assumption is often unreasonable, in particular for languages that use productive compounding to create novel word types. We therefore focus on modelling German since it makes extensive use of productive compounding and gives us an opportunity to explore this problem in depth. Second, in a naïve n-gram parametrisation, words are modelled independently of each other. This is problematic since the number of parameters is far too large to estimate reliably from even the largest corpora, and it ignores our intuition that related word forms have related behaviour. We solve these problems with an n-gram language model based on the hierarchical Pitman-Yor process (HPYP): Our model relaxes the closed vocabulary assumption by incorporating productive compound formation in its generative story, while the hierarchical structure enables us to relax the naïve independence assumptions about the statistical behavior of related word forms.

In the next section, we address the German compound problem in further detail and use this to motivate the structure of our model (§3). We then discuss the inference problem (§4), and evaluate the model's performance in terms of held-out perplexity and on translation quality when used inside an English→German translation system (§5). We conclude by placing this work in the context of related approaches (§6) and addressing avenues for future work.

2 Compound Words

Our aim in this work is to develop a language model that accounts for the structure of compound words. Compounding is a process whereby words are formed by combining other words. In some languages (including German, Swedish, Dutch and Afrikaans), compounds are written as single orthographic units. NLP systems that rely on whitespace to demarcate their elementary modelling units, e.g. the "grams" in n-gram models, are thus prone to suffer from sparse data effects that can be attributed to compounds specifically. An account of compounds in terms of their components therefore holds the potential of improving the performance of such systems.

Examples of compounds

- A basic noun-noun compound:
  Auto + Unfall = Autounfall *(car crash)*

[1] Preliminary work on the approach we follow in this paper was previously reported on by Botha (2012). Here, we expand on the scale and depth of the empirical evaluation and investigate an additional inference technique.

- *Linking elements* can appear between components
  Küche + Tisch = Küchentisch *(kitchen table)*
- Components can undergo stemming
  Schule + Hof = Schulhof *(schoolyard)*
- Compounding is recursive
  (Geburt + Tag) + Kind = Geburtstag + Kind = Geburtstagskind *(birthday boy/girl)*
- Compounding extends beyond noun components
  Zwei-Euro-Münze *(two Euro coin)* Fahrzeug *(vehicle)*

A compound is said to consist of a *head* component and one or more *modifier components*, with optional *linking elements* between consecutive components (Goldsmith and Reutter, 1998). The linguistic intuition that we propose to exploit in our language model is that German compounds are overwhelmingly right-headed (Toman, 1992), i.e. the right-most component fully determines the word's morphosyntactic properties. For example, the "Bahn" in "Eisenbahn" (railway) identifies the word as singular feminine, which determines the requirements for its agreement with verbs, articles and adjectives.

A language model could therefore give a reasonable assessment of the syntactic fluency of a sequence of German words by ignoring the non-head components of compounds. For example, the sentence, "I'm going by train" can be rendered in German as either of the following:

- Ich fahre mit der Eisenbahn.
- Ich fahre mit der Bahn.

Collapsing all compounds to their heads and ignoring modifiers would decrease sparsity and allow more robust n-gram probabilities to be estimated from data. But such a strategy would not be probabilistically sound as a generative model of a corpus. Moreover, a model that ignores modifiers would assign the same probability value to "Eisenbahn" and the empirically much rarer "Bobbahn" (bobsled), which would be unsatisfactory in a task where the language model plays a disciminative role. The model needs to account for the non-head components in some way. We expect the identity and number of modifier components to be strongly correlated with the identity of the head. In particular, the conditional distributions of modifier given head will be sharply peaked. A simple approximation is thus to assume that, conditioned on the head, modifiers are generated by a reverse n-gram model:[2]

$$p(\text{eisenbahn} \mid \text{mit der}) \equiv p(\text{bahn} \mid \text{mit der}) \times p(\text{eisen} \mid \text{bahn}) \times p(\$ \mid \text{eisen})$$

The sentinel \$ indicates the word boundary and doubles as a control on the number of modifiers. In general, we will use this process as a *back-off* strategy, i.e., when the trigram "mit der Eisenbahn" is unobserved. Note that this is markedly different from linguistically naïve back-off models that would score the unobserved trigram "mit der Eisenbahn" by falling back on bigram or unigram estimates. In our model, we instead permit the model to back off to this decomposition before dropping valuable context information.

3 An n-gram Model with Compounding

In this section we aim to marry an n-gram model with the intuition of compound formation that we proposed before. We present an extension of the hierarchical Pitman-Yor language model

[2]The majority of compounds have two components and thus match this assumption well enough. Multipart compounds where the modifiers themselves are compounds may violate it.

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. (Literally, "with the cable car"; idiomatically, "by cable car")

(HPYLM) (Teh, 2006) that fulfils this aim. The particular properties of the Pitman-Yor process (PYP) (Pitman and Yor, 1997) that we exploit are its flexibility to specify arbitrary back-off distributions (making it easy to incorporate an additional model) and the fact that it generates distributions that adapt well to power-law behaviour, as is often observed in language.

We employ this HPYLM framework with its accompanying inference machinery rather than a seemingly obvious alternative of using two distinct word-level and compound-level n-gram models. The reasons are that our unified model can learn a subtle interpolation between those levels, obviating the need to introduce and tune an extraneous interpolation scheme between sub-models, while opening the door for future extensions, e.g. analysing compounds occuring in the n-gram history.

3.1 Hierarchical Pitman-Yor Language Model (HPYLM)

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

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

where we 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 PYP priors 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).

The generative process for a word w in context $\mathbf{u}$ is:

$$\begin{aligned}
G_0 &= \text{Uniform}(|\mathscr{W}|) \\
G_\emptyset &\sim PY(d_0, \theta_0, G_0) \\
&\vdots \\
G_{\pi(\mathbf{u})} &\sim PY(d_{|\mathbf{u}|-1}, \theta_{|\mathbf{u}|-1}, G_{\pi\circ\pi(\mathbf{u})}) \\
G_{\mathbf{u}} &\sim PY(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \\
w &\sim G_{\mathbf{u}},
\end{aligned}$$

where $\pi(\mathbf{u})$ truncates the context $\mathbf{u}$ by dropping the left-most word in it. The hyperparameters are tied across all priors with the same context length $|\mathbf{u}|$. To explain this process in terms of the familiar trigram case, consider how the probability $p(w \mid u, v)$ comes to be. Let $\mathbf{u} = [u, v]$. $G_{[u,v]}$ is then the PYP-distributed distribution over w. The hierarchy arises by using as the base distribution for the prior of $G_{[u,v]}$ another PYP-distributed $G_{[v]}$, i.e. the distribution $p(w \mid v)$. 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 $\mathscr{W}$.

3.2 Hierarchical Pitman-Yor Language Model + Compounds (HPYLM+c)

We define a compound word $\tilde{w}$ as a sequence of components $[c_1, \ldots, c_z]$, 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 §2, we 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, we modify the HPYLM thus:

1. Replace the support with the reduced vocabulary $\mathscr{M}$, the set of unique elementary components c obtained when segmenting the items in $\mathscr{W}$. ($\mathscr{M}$ also includes items consisting of a single component to begin with.)
2. Add an additional level of conditional distributions $H_{\mathbf{u}}$ (with $|\mathbf{u}| = n-1$) where items from $\mathscr{M}$ combine to form the observed surface words.

The generative process changes as follows (see also Figure 2):

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

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:

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

The linguistically motivated choice for conditioning in F is $\Lambda^{\text{ling}} = [z, z-1, \ldots, 1]$ 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 $\Lambda^{\text{inv}} = [1, \ldots, z]$, where \$ this time marks the right word boundary and is drawn from $F(\cdot|c_z)$.

Linking Elements In the preceding definition of compound segmentation, the linking elements do not constitute items in the vocabulary $\mathscr{M}$. Regarding linking elements as components in their own right would sacrifice important contextual information and disrupt the conditionals $F(\cdot|c_{\Lambda_{i-1}})$. That is, faced with the compound Küche·n·tisch, we want P(küche|tisch) in the model, but not P(küche|n).

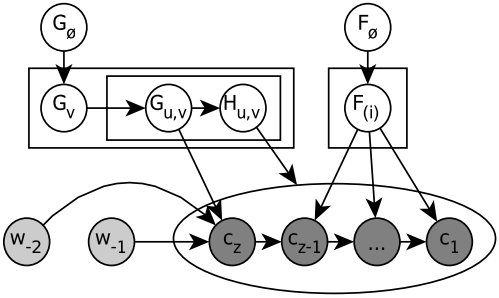

Figure 2: Plate diagram showing how a trigram version of HPYLM+c, using a bigram model F with condition scheme Λ^{ling} for modifiers, generates a word (the ellipse), consisting of head c_z and modifiers $c_1 \ldots c_{z-1}$. Here, w_{-2} and w_{-1} form the trigram context. We omit hyperparameters and their priors for clarity.

But linking elements must be accounted for to have a well-defined generative model. We follow the pragmatic option[3] of merging any linking elements onto the adjacent component – for Λ^{ling} merging happens onto the preceding component (e.g. P(küchen|tisch)), while for Λ^{inv} it is onto the succeeding one (e.g.P(ntisch|küche)). This keeps the 'head' component c_{Λ_1} intact.

4 Training

For ease of exposition we 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 $\mathscr{L} = (G_{[\emptyset]}, G_{[v]}, G_{[u,v]}, H_{[u,v]}, F_{\emptyset}, F_c)$, where $u, v \in \mathscr{W}$, $c \in \mathscr{M}$, and hyperparameters $\Omega = (d_i, \theta_i, d'_j, \theta'_j, d''_2, \theta''_2)$, 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 the latent variables in $\mathscr{L}$, 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 stick-breaking construction (Pitman, 2002b). But the more indirect representation using the Chinese Restaurant Process (CRP) (Aldous, 1985; Pitman, 2002a) 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 (type) w_i of

[3]It is worth noting that for German the presence and identity of linking elements between c_i and c_{i+1} are in fact governed by the preceding component c_i (Goldsmith and Reutter, 1998).

the corresponding token, and the way in which tables and dishes are chosen gives 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 for the $T + 1$ choices:

$$\Pr(k = k' | \ldots) \propto \begin{cases} C_{k'} - d & \text{occupied table } k' \in [1, T] \\ \theta + dT & \text{unoccupied table } k' = T + 1 \end{cases} \tag{2}$$

Ordering a dish for a new table corresponds to drawing a value ϕ_k from the base distribution G_0, and it is admissible 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

$$\Pr(w | \ldots) \propto C_w - dT_w + (\theta + dT)G_0(w), \tag{3}$$

where C_w is the number of customers of type w and T_w the number of tables serving dish w in the restaurant. 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).

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 spawned and sent to join the parent restaurant $\text{pa}(R)$. 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 $\text{pa}(R)$.

We take care to satisfy this constraint in our model where some restaurants have as base distribution a product of models. Here, when a new table serves a dish $\phi = \hat{w}$ in trigram restaurant $H_{[u,v]}$, a customer c_{Λ_1} joins the corresponding bigram restaurant $G_{[u,v]}$, *and* customers $c_{\Lambda_2}, \ldots, c_{\Lambda_z}, \$$ are sent to the restaurants for $F(\cdot|c_{\Lambda_1}), \ldots, F(\cdot|c_{\Lambda_z})$, respectively.

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 marginalised to compute the true posterior predictive probability of a word:

$$p(w|\mathscr{D}) = \int_{S,\Omega} p(w|S,\Omega)p(S,\Omega|\mathscr{D})\, dS\, d\Omega, \tag{4}$$

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, we place vague priors over them and use slice sampling[4] (Neal, 2003) to update their values during generation of the posterior samples: $d \sim \text{Beta}(1,1)$; $\theta \sim \text{Gamma}(10, 0.1)$

Lastly, we make the pragmatic approximation that $m = 1$, i.e. predictive probabilities are informed by a single sample[5] (S, Ω), taken after $B > 1$ iterations of burn-in.

Approximate Inference A common criticism of models like ours is that MCMC sampling increases training time unreasonably for an MT pipeline, despite the simplicity of Gibbs sampling.

To address this concern, we will evaluate the viability of using approximate inference in our model, inspired by the interpretation of original interpolated Kneser-Ney smoothing as approximate inference in the HPYLM (Teh, 2006; Goldwater et al., 2006). In each CRP, we constrain all the customers of a type to be seated at the same table, $T_w = 1 \; \forall w$. This changes the predictive probability of a word to

$$\Pr(w|\ldots) \propto C_w - d + (\theta + dT')G_0(w), \tag{5}$$

where T' is now the number of unique types in the restaurant. In the hierarchical model, this implies absolute discounting of the n-gram counts by an amount d.

Under this scheme, the calculation of all C_w and T' across the hierarchy is deterministic. We can therefore obtain the full seating arrangement S from a single pass through the training data. We update the hyperparameters as described in the previous section, although an alternative would be to tune them against perplexity on development data.

5 Experiments

In this section we report on experiments performed to gain insight into the behaviour of the proposed model. The first task we evaluate on is the model's ability to predict a previously unseen text. Our aim is to establish whether the model's account of compounds benefits it without hampering its global performance. We also investigate how the performance depends on the amount of context used when predicting tokens, and on the amount of training data used to estimate the model.

Secondly, we are interested in how the model interacts with a large-scale statistical machine translation system when translating from English to German. Compound words are known to be a challenging aspect of this task, and the aim is to see if specifically accounting for them in the language model can bias a decoder towards better translations. We did not modify other aspects of the translation system, thus it cannot hypothesise "new" compounds and will not benefit from our model's ability to score unseen compounds consisting of observed components.

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

[5] Our preliminary experiments indicated that the posterior over the latent model structure is quite sharply peaked, so that a single sample constitutes a low-variance estimator of the posterior predictive distribution.

5.1 Methods

Data and Tools All data we used are from the WMT11 shared-task.[6] Standard data preprocessing steps comprised normalising punctuation, tokenising and lowercasing all words.

For language model training, we used the union of the news commentary data, Europarl and the news article corpus for 2011. Preprocessing and deduplication yielded a corpus of 59m running tokens, roughly a fifth of all the German monolingual data supplied in WMT11 when using the same preprocessing. No pruning was done on the n-gram counts, but we mapped training tokens to the "unknown" token if they do not appear in the target-side of the bitext (see below). The motivation is that the hypotheses to be scored against the language model during decoding are by definition constrained to this vocabulary.

Our test corpus for the monolingual task is the union of all the WMT11 development data for German (`news-test2008,9,10`, 7065 sentences).

For translation experiments, the preprocessed English-German bitext was filtered to exclude sentences longer than 50 tokens, resulting in 1.7 million parallel sentences; word alignments were inferred from this using the Berkeley Aligner (Liang et al., 2006) and used as a basis from which to extract a Hiero-style synchronous CFG (Chiang, 2007).

The weights of the linear translation models were tuned towards BLEU using `cdec`'s (Dyer et al., 2010) implementation of MERT (Och, 2003). For this, the development set `news-test2008` (2051 sentences) was used, while final BLEU scores are measured on the official test set `newstest2011` (3003 sentences, 171460 tokens), without detokenising or recasing hypotheses.

Compound segmentation For this evaluation, we used an *a priori* segmentation of compounds into parts to build our models. This means we assume a single, fixed analysis of a compound regardless of the context it occurs in, which is necessitated by the fact that our probabilistic model does not specify a step for choosing an analysis. To construct a segmentation dictionary, we ran a supervised[7] compound splitter (Dyer, 2009) on all the words[8] in the training vocabulary, retaining the one-best segmentation. In addition, word-internal hyphens were also taken as segmentation points. Finally, linking elements were merged onto components as discussed in §3.2. Any token that is split into more than one part by this procedure is regarded as a compound, and we find that the majority of compounds thus identified consist of one or two parts (Table 1b).

5.2 Compounds as n-grams

Our model is premised on the idea that better probability estimates can be obtained by analysing compounds into their components. To investigate this claim empirically, we trained a variety of 4-gram language models and compare them by how well they predict an unseen text consisting of N tokens. For each model q, we report measurements in terms of perplexity, $\text{PPL}=\exp\left(-1/N \sum_\tau \ln q(\tau)\right)$, calculated over all tokens τ in the text.

It should be noted that the domain of our model is a countably infinite set. According to the generative process of HPYLM+c (§3.2), there is no theoretical limit on the number of parts in a

[6] http://www.statmt.org/wmt11/

[7] We chose a supervised splitter as the focus of our evaluation is on the language model's subsequent use of the segmentation, not on the quality of the segmentation itself. Unsupervised methods could also be used with our model.

[8] We also included tokens having numerals and at least two letters, e.g. "CO2-handel" (carbon trade)

	En	De	De LM
Sentences	1.7m	1.7m	2.4m
Tokens	49m	38m	59m
Token Types	112k	351k	596k

(a) Statistics of training corpora.

Parts per Compound	Compound Types
2	197233
3	25128
4	1194
≥5	59

(b) Compound types by length.

Table 1: Summary of training data and compound segmentation.

compound; there is always a non-zero probability of adding another modifier c from $\mathscr{M}$ to a partially formed compound. In this evaluation, we used the probabilities supplied by HPYLM+c without normalising over the finite vocabulary $\mathscr{W}$. Consequently, a comparison to baseline models that have a finite domain is somewhat biased in their favour.

Our main model of interest is HPYLM+c using the Λ^{ling} segmentation and a bigram model F_{HPYLM} over modifiers. To measure the importance of adhering to linguistic intuition, we also evaluate the variant using Λ^{inv}, other things equal. As baselines we used an interpolated, modified Kneser-Ney model (mKN) and an HPYLM. For the sampling-based models, we took one sample from the posterior after $B = 300$ iterations of burn-in.

We find that the main model achieves a slightly lower perplexity than HPYLM, which in turn beats the mKN baseline by 1.9% (Table 2a). The use of the linguistically implausible scheme Λ^{inv} has a noticeably detrimental effect on performance.

	Perplexity
mKN	299.9
HPYLM	294.1
F_{HPYLM} Λ^{ling}	293.6
F_{HPYLM} Λ^{inv}	305.5

(a) Performance of 4-gram models against baselines. Lower is better.

	n=2	**n=3**	**n=4**
mKN	394.5	307.2	299.9
HPYLM	396.6	303.3	294.1
F_{HPYLM} Λ^{ling}	390.0	299.3	293.6

(b) Test-set perplexity for different n-gram orders.

Table 2: Comparison of language models and effect of n-gram order.

For a more qualitative insight into the model performance, we did a further direct comparison of our main model and the mKN baseline by ranking test set compounds by the difference in probability value that each model assigns to the n-gram. The test compounds where the compound model does best (Table 3 top) are all words for which an analysis into a context-dependent head and modifiers should clearly be beneficial. For example, in scoring the phrases "wochen vor den präsidentschaftswahlen" (weeks before the presidential elections) and "tage vor den parlamentswahlen" (days before the parliamentary elections), the head "wahlen" is having a mutually reinforcing effect. In contrast, we find that the cases where the mKN baseline model does best (Table 3 bottom) feature various words that are not strictly speaking compounds, but largely artefacts of our segmentation method: e.g. mistakes such as "ging+rich" or "wissen+schaften", or greediness from splitting on hyphens, e.g. "ki-+moon". These are words where our compound model's smoothing is hurting performance, since it allocates some

HPYLM+c better	Δ
gegen die umstrittene **wieder+wahl**	0.058
aufbau der afghanischen **sicherheits+kräfte**	0.036
dessen zentralen **gesichts+punkten**	0.035
in annapolis , **mary+land**	0.035
wochen vor den **präsidentschafts+wahlen**	0.032
dieses vertrauen nicht **miss+brauchen**	0.030
für psychiatrie und **psycho+therapie**	0.028
tage vor den **parlaments+wahlen**	0.028
reduktion der **treibhausgas+emissionen**	0.025
in einem unblutigen **militär+putsch**	0.021
Baseline (mKN) better	**Δ**
, newt **ging+rich**	0.511
nächtlichem **flug+lärm**	0.449
generalsekretär ban **ki-+moon**	0.423
in st. **peters+burg**	0.420
im 17. **jahr+hundert**	0.419
saalpublikums in st. **peters+burg**	0.359
militanten klerikers moqtada **al-+sadr**	0.352
un-hochkommissarin für **menschen+rechte**	0.286
schwebt in **lebens+gefahr**	0.231
der akademie der **wissen+schaften**	0.212

Table 3: Compounds from the monolingual test set for which HPYLM+c outperforms mKN by the largest margin (top) and vice-versa (bottom). We define the margin Δ as the difference in probability that the models assign to the given test n-gram.

probability mass toward observing other modifiers with the head, which in the case of these proper nouns will not happen. This is evidence of success on the part of our model's underlying mechanism, but demonstrates that more care should be taken with the particular segmentation method used.

5.3 Scaling

Here we consider the behaviour of our model under scaling along two dimensions: n-gram order and training data size.

Our model reduces data sparsity by generalising over different compounds that have the same head. But this happens at the maximal n-gram order, meaning the full surface form is not available in the lower-order conditional distributions. There may be cases where this amounts to "premature back-off" when the lower-order distributions are very informative for a particular surface form.

To see if this has an observable effect, we performed an additional experiment using orders $n = 2$ and $n = 3$. The results in Table 2b indicate that we maintain a lower perplexity than the baselines.

For $n = 2$ and $n = 3$, the sampler had not fully converged after 300 iterations. We suspect this is due to the higher entropy in the distributions governing the seating assignments: If $n = 2$, there should be more customers (and therefore more seating configurations) in the

average restaurant for context-length 2 than in the same restaurant if n is larger. This did not affect perplexity, which was stable when evaluating with different individual samples from the posterior around 300 iterations.

The other dimension of scaling is training data size. We drew random subsamples of different sizes from our training corpus for training further language models.

For small data sizes, the baseline models achieve a noticeably lower perplexity than our compound model. This is contrary to the effect we expected in light of the sparsity reduction our model brings. We suspect that this is primarily due to the lack of normalising the model over a finite vocabulary. For larger sizes, it is competitive against the baselines once more.

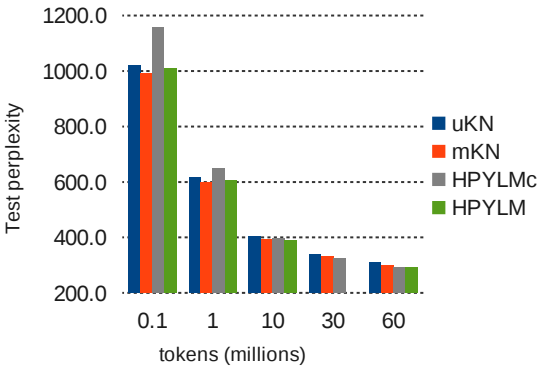

Figure 3: Test perplexity for different sizes of training data, keeping $n = 4$ fixed. uKN and mKN are original and modified Kneser-Ney, respectively, both using interpolation.

5.4 Effect on Translation

The performance of an n-gram language model in an intrinsic evaluation does not necessarily correlate with the effect it has when used as part of a translation system. We thus conducted a separate translation experiment, comparing the quality of the output produced by the translation system described in §5.1 when using different language models.

In terms of BLEU score, we do not find a meaningful difference between the various systems (Table 4a). The system using our main model matches the two baselines, a result that indicates our more expressive modelling is not sacrificing any performance in this task. This is an important outcome, as it means we avoid a common pitfall whereby a new model is proposed to target some specific phenomenon, does so successfully but then sacrifices performance globally. The linguistically implausible segmentation scheme again performs slightly worse.

When using the approximate inference scheme, denoted by `1tbl` in these results, we firstly find that language model training reduces to a trivial amount of time compared to the proper samplers; for `1tbl`, the posterior likelihood converged fully within 5 iterations, where an iteration comprises merely resampling the hyperparameters. By contrast, the posterior likelihood

	PPL	BLEU
mKN	299.9	13.9
HPYLM	294.1	13.9
F_{HPYLM}, Λ^{ling}	293.6	13.9
F_{HPYLM}, Λ^{inv}	305.5	13.7
F_{HPYLM}, Λ^{ling} `1tbl`	355.4	13.6

(a) BLEU over 3003 test sentences with single references. The standard deviation in BLEU score across the three independent runs varied between 0.1 and 0.3. For reference, we also show the perplexity of each model on the monolingual test set.

	P	R	F
mKN	25.4	17.1	20.5
HPYLM	24.3	**17.5**	20.4
F_{HPYLM}, Λ^{ling}	**27.5**	17.3	**21.3**
F_{HPYLM}, Λ^{inv}	23.7	17.2	19.9

(b) Precision, Recall and F-score for compounds in the translation output, relative to the reference set containing 2652 compounds. Each value is calculated across the union of hypotheses produced by decoding the test set with the weights obtained from the three independent runs.

Table 4: Translation results over three MERT runs, using 4-gram language models.

under proper sampling was still improving marginally after 300 iterations for the other models, where one iteration comprises hyperparameter resampling *and* a pass through all training tokens to resample their seating assignments in the CRP.

The `1tbl` model achieved a worse perplexity in the monolingual evaluation task, but with only a small negative effect on BLEU score compared to the baseline (Table 4a). This result suggests there is some leeway in the development of models in the HPYLM framework to explore in future work: model complexity can be pushed up by trading off predictive accuracy against training time.

Next, we turn to a more fine-grained look at the translation output. The BLEU metric is likely to miss small improvements in translation quality. Moreover, in our test corpus only 2652 of the 72661 reference tokens are compounds; a moderate improvement in generating them is unlikely to have a big impact on the BLEU.

To establish whether the model aids in the translation of compounds in particular, we measured the accuracy of hypotheses produced by the different translation systems against the reference translations. We use the standard metrics of precision (correct compounds as a fraction of all compounds output) and recall (correct compounds as a fraction of the compounds in the references).

The results in Table 4b show that using our model increases compound precision by 12% against the HPYLM baseline and 8% against the Kneser-Ney baseline (relative increases). The fact that recall remains stable proves that the gain in precision is not achieved simply by the system being more conservative about outputting compounds in the first place.

6 Related Work

Bilmes and Kirchhoff (2003) proposed a more general framework for n-gram language modelling, which can also be used for implementing sparsity reduction measures. Their Factored Language Model (FLM) views a word as a vector of features, such that a particular feature value is generated conditioned on some history of preceding feature values. This allows one to construct n-gram models with dependencies among sequences of PoS tags or semantic classes in addition to standard word-based dependencies. It should be possible to encode a model with structure comparable to ours in the FLM framework, but it does not lend itself naturally to

having a variable number of features depending on the predicted token in the way our model allows a variable number of parts in a compound.

Another common approach for addressing the sparsity effects of compounding (Koehn and Knight, 2003; Koehn et al., 2008; Stymne, 2009; Berton et al., 1996), and rich morphology (Habash and Sadat, 2006; Geutner, 1995), has been to use pre/post-processing with an otherwise unmodified translation system or speech recognition system. This approach views the existing machinery as adequate and shifts the focus to finding a more appropriate segmentation of words into tokens, i.e. compounds into parts or words into morphemes, thus achieving a vocabulary reduction. The downside of such a method is that training a standard n-gram language model on pre-segmented data introduces unwanted effects: in the case of German compounds, the split-off modifiers would take precedence in a split-off head's n-gram context, and during back-off the actual word-context information is discarded first. The problem is similar when modelling sequences of morphemes as n-grams, and earlier work in speech recognition has shown that taking steps against this effect can improve recognition accuracy (Ircing et al., 2001). Pre-processing also often requires heuristics to guard against over/under-segmentation, which do not generalise well to different settings or languages.

Our work is also subject to the whims of our compound segmentation method, but the model is more robust since it does retain the original surface form of the word – recall that the decomposition step amounts to interpolated back-off.

Baroni and Matiasek (2002) proposed basic models of German compounds for use in predictive text input, exploiting the same link between right-headedness and context as we have, although their focus was restricted to compounds with two components.

In terms of Bayesian modelling, the PYP has been found to be very useful in a variety of tasks, including word segmentation, speech recognition, domain adaption and unsupervised PoS tagging (Goldwater et al., 2006; Mochihashi et al., 2009; Huang and Renals, 2007; Neubig et al., 2010; Wood and Teh, 2009; Blunsom and Cohn, 2011). In all cases its power-law scaling and ease of extensibility via the base distribution allowed the formulation of interesting models that achieved competitive results.

7 Conclusion

We have demonstrated how an existing hierarchical Bayesian model can be used to build an n-gram language model that is informed by intuitions about the specific linguistic phenomenon of closed-form compounds. While our focus was on compounds, we argue that this approach can be useful for other phenomena, such as rich morphology more generally, where data sparsity creates smoothing problems for n-gram language models.

Our empirical results support the conclusions that the increased model expressiveness has a positive impact on the monolingual task of predicting unseen German text, outperforming a competitive Kneser-Ney baseline. When used as part of an English→German translation system, there was little effect on the BLEU metric, but the model was associated with an increase in the F-score for generating correct compounds during translation.

Future work will entail extending the translation system to hypothesise novel compounds, a situation where a productive language model should be vital for generating fluent translations. Further modelling work is therefore needed to handle novel compounds that occur in the n-gram history.

Acknowledgements

We thank the anonymous reviewers for their feedback, and acknowledge the support of The Rhodes Trust (Botha), EPSRC grant number EP/I010858/1 (Blunsom) and the U. S. Army Research Office under contract/grant number W911NF-10-1-0533 (Dyer).

References

Aldous, D. (1985). Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII — 1983*, volume 1117 of *Lecture Notes in Mathematics*, pages 1–198. Springer.

Baroni, M. and Matiasek, J. (2002). Predicting the components of German nominal compounds. In *Proc. of the European Conference on Artificial Intelligence (ECAI)*, pages 470–474.

Berton, A., Fetter, P., and Regel-Brietzmann, P. (1996). Compound Words in Large-Vocabulary German Speech Recognition Systems. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 1165–1168. IEEE.

Bilmes, J. A. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *Proc. of NAACL-HLT (short papers)*, pages 4–6. Association for Computational Linguistics.

Blunsom, P. and Cohn, T. (2011). A Hierarchical Pitman-Yor Process HMM for Unsupervised Part of Speech Induction. In *Proc. of ACL*. Association for Computational Linguistics.

Botha, J. A. (2012). Hierarchical Bayesian Language Modelling for the Linguistically Informed. In *Proc. of the EACL Student Research Workshop*, pages 64–73.

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

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

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). cdec: A Decoder, Alignment, and Learning framework for finite-state and context-free translation models. In *Proc. of ACL (Demonstration session)*, pages 7–12. Association for Computational Linguistics.

Geutner, P. (1995). Using morphology towards better large-vocabulary speech recognition systems. In *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 445–448.

Goldsmith, J. and Reutter, T. (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.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2006). Interpolating Between Types and Tokens by Estimating Power-Law Generators. In *Advances in Neural Information Processing Systems, Volume 18*.

Habash, N. and Sadat, F. (2006). Arabic preprocessing schemes for statistical machine translation. In *Proc. of NAACL-HLT*, pages 49–52. Association for Computational Linguistics.

Huang, S. and Renals, S. (2007). Hierarchical Pitman-Yor Language Models For ASR in Meetings. In *Proc. of Workshop on Automatic Speech Recognition and Understanding*, pages 124–129. IEEE.

Ircing, P., Krbec, P., Hajič, J., Psutka, J., Khudanpur, S., Jelinek, F., and Byrne, W. (2001). On large vocabulary continuous speech recognition of highly inflectional language - Czech. In *Proc. of Interspeech*, pages 487–490.

Koehn, P., Arun, A., and Hoang, H. (2008). Towards better Machine Translation Quality for the German – English Language Pairs. In *Proc. of Workshop on Statistical Machine Translation*, pages 139–142. Association for Computational Linguistics.

Koehn, P. and Knight, K. (2003). Empirical Methods for Compound Splitting. In *Proc. of EACL*, pages 187–193. Association for Computational Linguistics.

Liang, P., Taskar, B., and Klein, D. (2006). Alignment by Agreement. In *Proc. of NAACL-HLT*, pages 104–111. Association for Computational Linguistics.

Mochihashi, D., Yamada, T., and Ueda, N. (2009). Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proc. of ACL-IJCNLP*, pages 100–108, Suntec, Singapore. Association for Computational Linguistics.

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

Neubig, G., Mimura, M., Mori, S., and Kawahara, T. (2010). Learning a Language Model from Continuous Speech. In *Proc. of Interspeech*, pages 1053–1056, Chiba, Japan.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL*, pages 160–167.

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

Pitman, J. (2002b). Poisson-Dirichlet and GEM Invariant Distributions for Split-and-Merge Transformations of an Interval Partition. *Combinatorics, Probability and Computing*, 11(05):501–514.

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

Stymne, S. (2009). A comparison of merging strategies for translation of German compounds. In *Proc. of EACL Student Research Workshop*, pages 61–69. Association for Computational Linguistics.

Teh, Y. W. (2006). A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. of ACL*, pages 985–992. Association for Computational Linguistics.

Toman, J. (1992). Compound. In Bright, W., editor, *International Encyclopedia of Linguistics*, volume 1, pages 286–288. Oxford University Press.

Wood, F. and Teh, Y. W. (2009). A Hierarchical Nonparametric Bayesian Approach to Statistical Language Model Domain Adaptation. In *Proc. of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 607–614, Clearwater Beach, Florida, USA.