Enlightening the Bulb: Unsupervised learning of morphology for word and subword alignments

Pierre Godard François Yvon
LIMSI, CNRS, Université Paris-Saclay
prenom.nom@limsi.fr

Abstract

There is an abundant literature elaborating on the topic of the unsupervised learning of morphological structures in language, mostly concerned with the segmentation of a text into words or morphemes, and the elaboration of morphological paradigms. Likewise, much work has been undertaken since more than twenty years in order to compute alignments at the word level and without supervision, given a parallel corpus. Attempts to combine both approaches into a unified framework, able to learn morphology and alignment structures jointly, appear to be much less numerous and correspond to a more recent trend of research. In the context of the BULB project, aimed at documenting unwritten and endangered languages, we are interested in methods that can make up for the lack of ressources with the additional information contained in parallel data. We provide in this report a global review of this literature.

1 General introduction

The main objective of BULB [1] is to support the documentation of unwritten and endangered languages. For the course of the project, the main focus will be three mostly unwritten African languages of the Bantu family (Basaa, Myene and Embosi). The data collection scenario is the following and mostly derives from proposals initially made in (Bird, 2011; Bird and Chiang, 2012): a corpus of speech in the unwritten language (henceforth UL) is collected, and possibly recorded a second time through re-speaking so as to provide a cleaner, more articulated signal. This corpus is then orally translated into a well-resourced language (here: French). Automatic Speech Recognition (ASR) subsequently provides a phonetic transcription of the UL recordings as well as a word transcription of the oral French translations. From there, cross-lingual alignment should be automatically performed between UL phone sequences and French words, providing automatic annotations that should help linguists in their study and documentation of the UL.

[1] BULB is an ANR/DFG project funded under the 2014 call.

This document focusses on currently available methods allowing for the learning, in an unsupervised manner, of such multi-level (phones to words) cross-lingual alignments. This requires to study, first, the unsupervised learning of morphology in a monolingual context and, second, automatic word alignments in a standard bilingual set-up. In a third section, we will detail existing joint models able to learn both segmentation and alignment. A presentation of the most complex mathematical formalisms can be found in the appendix.

2 Unsupervised Learning of Morphology

2.1 Introduction

In this first section, we focus on the *monolingual aspects* of the task at hand, and provide an account of recent works aimed at learning a (weak) theory of morphology in an unsupervised manner. Our scope is even slightly more general, as we will also mention works aimed at learning a lexicon from raw, unsegmented text (or speech). This is because, from an abstract point of view, morphology learning and lexical acquisition problems can be viewed as instances of a same generic task, which is to learn to segment an input stream of symbols in an unsupervised way and to extract a minimal inventory of units, be they called words or morphemes. A third related trend of research should also be mentioned here: the unsupervised segmentation of sentences in words for languages having no overt word separator in their orthography (Chinese, Japanese, Thai, etc.).

Hammarström and Borin (2011) present a high-level comprehensive survey of Unsupervised Learning of Morphology (ULM), and provide a quite useful entry point to this large body of literature. It clearly distinguishes between two quite separated trends:

- to provide with an operational description of morphological phenomena (usually restricted to rather simplistic forms of concatenative processes). The input is here typically a set of raw word forms that need to be explained in the most economic way possible - for instance using the Minimum Description Length (Rissanen, 1989);
- to provide models for language acquisition; this is a much smaller body of literature, which also tries to consider supplementary information to raw (phoneme) strings: intonation, but also semantics, pragmatics etc.

Another merit of this non-technical paper is to try to be explicit about (a) the theory of morphology that is underlying the various statistical models used for ULM (mostly in the line of “Items and Arrangements”); and (b) the output of the morphology learning system: a segmentation machine; an inventory of morphemes; paradigms; a semantic interpretation of morphemes, etc.

One needs to recall that the diversity of approaches to the unsupervised learning of morphology is largely induced by the broad variety of natural languages. We borrow from linguistic typology (Eifring and Theil, 2004) the distinction between *analytic*, *synthetic* and *polysynthetic* languages.[2] The latter two classes of languages may also

[2] Analytic, or isolating, languages tend to feature words corresponding to only one morpheme, whereas

be further divided into so-called *agglutinative* and *flective* languages.[3]

The upcoming section will unravel two main different trends of research. A first trend looks at morphology learning as a segmentation task using context, particularly suited for analytic or agglutinative languages which tend to exhibit a one-to-one correspondence between meaning and form. A second trend will, on the contrary, attempt to model the relations between forms, and build paradigmatic structures from those relations; this last trend will likely better accomodate the morphological behavior of flective languages. We will also show that some works, for example the work of Goldsmith (2001), lie between these two poles, involving a concept of *signature* akin to a paradigm yet relying mostly on a segmenting approach.

2.2 Early models for unsupervised string segmentation

Harris (1955) pioneered automatic morphology discovery, observing that transitions between morphemes inside a word are less predictable than transitions between phonemes within a morpheme. Counting the number of phonemes that could extend any prefix (resp. suffix) into another legal prefix (resp. suffix) in the language – the *successor* (resp. *predecessor*) *frequency* – it is possible to introduce without any supervision a boundary, within a word, at positions that correspond to peaks of that frequency. This approach, and its information-theoretic interpretations using mutual information and entropy measures, proved to be extremely influential. Déjean (1998), in particular, devised an unsupervised morpheme discovery procedure using Harris' local association statistics during a bootstrapping step, subsequently expanding the morpheme list with morphemes appearing in similar contexts to the ones already discovered.

Deligne et al. (2001) expose exhaustively different concepts introduced through a series of papers addressing the problem of segmenting one or multiple streams of symbols in a non-supervised manner. Deligne and Bimbot (1995) present the "multigram model" in which a sentence (or more generally a stream of symbols) is seen as the concatenation of independent sequences of words (resp. symbols). Those sequences' length can vary up to a maximum length n. One way to look at the n-multigram model is to think of it as a n-state Hidden Markov Model with state i emitting only sequences of length i and all transition probabilities being equal (see Figure 1). This allows for an efficient training of the model using the EM algorithm and a forward-backward procedure so as to avoid enumerating all the possible segmentations.

This forward-backward procedure is slightly modified from the standard approach used for HMM training, in order to take into account the dependency of the number of emissions with respect to the segmentation. The multigram model is compared successfully, as a language model, to n-gram models of different order. It is also subsequently implemented in the context of the unsupervised segmentation of phoneme strings (Bimbot et al., 1995).

in synthetic (resp. polysynthetic) languages, words correspond to more than one morpheme (resp. several morphemes constituting sometimes up to an entire clause).

[3] Agglutinative languages have the following properties: a) they make use of morphemes that express only one meaning element, b) their morphemes have clear boundaries, and c) grammatical processes adjoining prefixes and suffixes do not modify morphemes' forms. Conversely, flective languages display the opposite properties, called *cumulation*, *fusion* and *introflection*.

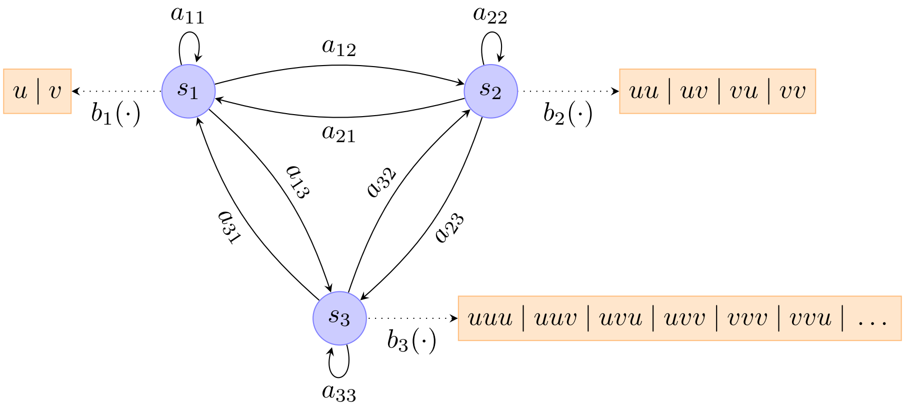


Figure 1 – A HMM corresponding to a 3-multigram model with 3 states s_1, s_2, s_3 which can emit respectively 1, 2 or 3 symbols from the vocabulary $\{u, v\}$. The transition probabilities a_{ij} from state s_i to state s_j are uniform; $b_i(x)$ represents the probability to emit symbol x in state s_i.

The model is then extended in (Deligne et al., 1995) to a *joint* version which is able to deal with two or more streams of symbols, which are themselves seen as different transcodings of the same higher level symbol stream. To make the model tractable, it is again necessary to limit the maximum length of units in both streams. A n, m joint multigram model can be then identified to a HMM where each of the $n \times m$ states emits pairs of i symbols in one stream, and j symbols in the other. This new joint multigram model is able to learn joint segmentations through many-to-many sequential pairings. The authors remark that when using this model to learn graphemic and phonetic pairings from words, the extracted joint units often have a morphological interpretation. Finally, Deligne and Bimbot (1997) investigate the particular problem of speech recognition through the lenses of the multigram model, with the ambition to reduce the assumptions usually made on the number and length of acoustic units paired with (also variable-length) phonetic sequences.

de Marcken (1996a) proposes that, if a lexicon is given, it is possible to learn a (locally optimal) parsing of the input data via the EM algorithm. In the case of a parsing that involves only concatenation and according to the work just mentioned from Deligne and Bimbot (1995), this corresponds to an implementation of the Baum-Welch algorithm (EM with a forward-backward procedure). From there, it is possible to infer probabilities for the lexicon, hence its codelength.[4] Building on the Minimum Description Length criterium (MDL) introduced by Rissanen (1989), the main idea consists in positing that, if a lexicon minimizes both its own description length (briefly, the space needed to encode it) and the description length of the data, then that lexicon is the theory that best explains the data and should be able to capture some of the principles at work in the language that originated the data.

The difficulty, as with most approaches we will encounter in the MDL framework

[4] The codelength of a word w with probability $p(w)$ is approximately $-\log p(w)$.

is that the model class (here the class of possible lexicons) can be very large, if not infinite, forcing those approaches to implement heuristics that depart strongly from the principled intention that initiated the work. de Marcken (1996b) elaborates with great details on this subject, and more generally on difficulties encountered in the study of Language Acquisition, introducing new computational methods to learn language structures in an unsupervised way, and showing the link between the MDL principle and a Bayesian view, in which MDL can be interpreted as a Bayesian prior biased against overly complex hypotheses.

2.3 Morfessor and the like

Described in an influential series of papers spanning over a decade, Morfessor[5] (and all its variants) has been established as a *de facto* standard for unsupervised morphology learning (UML), which is especially suitable for modeling concatenative morphology. In (Creutz and Lagus, 2002), a first basic model is described, where ULM is performed using again the Minimum Description Length (MDL) principle (Rissanen, 1989). The training objective is to minimize the size of the coding of the corpus, decomposed as (a) the size of the morph dictionary, computed as the sum of the morph lengths; and (b) the size of the corpus encoding when each morph m is coded with $-\log p(m)$ bits. The optimal codebook is obtained through heuristic (local) search. The corresponding generation process is a semi-Markov unigram model with a prior on the codeword lengths, which can also be estimated using conventional EM (the authors actually use Viterbi-style EM with various add-ons). As it turns out, the method that includes a prior on morphs performs slightly better than the pure ML estimation. This model is refined and extended in follow-up publications as follows. A minor change is first introduced in (Creutz, 2003), where the morph generation model is replaced by a unigram of characters, and where a much more complex prior on the morph codebook, integrating both length and type distribution,[6] is used. More substantial is the improvement in (Creutz and Lagus, 2004), which introduces some morphotactics and has distinct hidden states for prefix, stem and suffixes. The initialization of the emission distribution for these categories[7] is based on a preliminary segmentation (presumably using the simpler model of Creutz and Lagus (2002)), which, contrarily to Morfessor-Category, operates on lists of tokens rather than lists of types. This model is further refined in (Creutz and Lagus, 2005) (see also (Creutz and Lagus, 2007) for a more comprehensive presentation), where (a) segmentation can be performed recursively[8] and (b) a supplementary category is introduced to represent non-morph, so as to catch short strings that could be mistakenly interpreted as morphs (eg: *st* in *station*) – these non-morph segments are removed in a post-processing step, thereby reducing the tendency to oversegment of the original approaches. The introduction of recursive

[5] See `http://www.cis.hut.fi/projects/morpho` for open source implementations of Morfessor.

[6] The approach used to simulate power-law distribution used here is probably overtly complex - cf. the work with non-parametric Bayesian models below.

[7] That would otherwise be undistinguishable.

[8] meaning that eg. a suffix can itself be decomposed – think of a word such as *creations* where the suffix *-ations* can be decomposed as *-ation+s*.

segmentations also implies to revise the emission model of a morph, which, once fully decomposed, takes the form of a unigram of letters. Kohonen et al. (2010) is a first, and rather ad-hoc attempt to introduce annotated (ie. segmented) data in conjunction with non-annotated data, using the model of Creutz and Lagus (2005): as in most semi-supervised approaches, the objective function combines two likelihood terms that need to be carefully weighted. (Grönroos et al., 2014) is the latest evolution of Morfessor, where the authors try to better combine the benefits of semi-supervision (hence the use of a proper generative model) and of a richer morphotactics model (which distinguishes for morph types).

2.4 Towards Learning Paradigms

2.4.1 Linguistica

Goldsmith (2001) goes one step further than the identification of morpheme inventories and the learning of morphological segmentation procedures, and additionally studies the identification of *signatures*.[9] Signatures can be viewed as a weaker form of linguistic paradigms, consisting of sets of suffixes that systematically alternate with a set of stems (see Figure 2). Like for Morfessor (see Section 2.3), the approach is based on the Minimum Description Length, which is instantiated here as follows. The model is made of sets of stems, suffixes, and (the main novelty) *signatures*, which record the possibility that a stem and a suffix can actually cooccur. Denoting t a stem, f a suffix, w a word, and σ a signature, the probabilistic model which underlies the compression algorithm can be expressed as:

$$P(w = tf) = \sum_{\sigma} P(\sigma)P(t|\sigma)P(f|\sigma) \tag{1}$$

Equation (1) serves to compute the size of the data, given the model; the model size takes into account the length of the encoding of the lists of stems, suffixes, and signatures.[10] As discussed by the author, the introduction of signatures is essential to prevent the model to produce degenerate solutions.

$$\left\{\begin{array}{l}\text{jump}\\ \text{laugh}\\ \text{walk}\end{array}\right\}\left\{\begin{array}{l}\text{NULL}\\ \text{ed}\\ \text{ing}\\ \text{s}\end{array}\right\}$$

Figure 2 – An example of signature which covers the words *jump, jumped, jumping, jumps, laugh, laughed, laughing, laughs, walk, walked, walking, walks*.

It is noteworthy that the generative model for a word (in the simplest stem+suffix case– see Equation (1)) is almost similar to PLSA (Hofmann, 2001) or to other standard models for co-occurrences (see of eg. (Hofmann and Puzicha, 1998), which is a

[9] This short presentation can not do full justice to this work, which contains many subtle details and very enlightening comments regarding the qualitative behaviour of the learning algorithm.

[10] The actual coding scheme is somewhat generalized to account for the possibility to decompose a word in more than two morphs.

precursor of the PLSA paper, see also (Blitzer et al., 2005)). In such so-called aspect models, the joint probability of co-occurring events (here: suffix and stem) is explained by a latent variable (here the signature), conditioned on which the two parts are independent. This also provides a factorization of the conditional probability of the stem given the suffix using the latent signature (or the other way around) as: [11]

$$P(t|f) = \sum_{\sigma} P(t|\sigma)P(\sigma|f) \tag{2}$$

This observation may not be so helpful here as suffixes are not so many, meaning that (if it were observed) the corresponding matrix would be more rectangular than square. See the work by Chan (2006) below for further developments along these lines.

If the underlying principles behind Linguistica are well motivated and based on notions of text compression, the algorithmic part is more *ad hoc* and can be decomposed into the following steps: (a) a first decomposition into stem+suffix is computed (several methods are considered, such as building an inventory of frequent suffixes), which is used to compute a first list of signatures; (b) various local refinements of this initial solution are then explored in a greedy fashion, aimed at minimizing the overall description length.[12] Examples of such refinements are: to sub-decompose a suffix in two parts (eg. *+ments* decomposed as *+ment+s*) or to merge resembling signatures. Fine grained heuristics, the validity of which is discussed at length, are finally used to remove signatures having a too narrow support.

An interesting question is the completeness of signatures: for long signatures, eg. a complete verbal paradigm, many sub-signatures also exist in the data (corresponding to partial paradigms) - how can they be merged since there is no way to "hallucinate" forms that are not in the original list as this would not help to compress the data ? Here again, Goldsmith is pretty clear regarding the impact of this defect of the model, and tries (in Section 10 of (Goldsmith, 2001)) to provide some hints as to how this could be fixed.

Hu et al. (2005) further expand this trend of research, interpreting Goldsmith's signature as a finite-state automaton (FSA) (see Figure 3) built from character-based alignments between pairs of word forms. These alignments are established using the string edit distance (SED), identifying perfect and imperfect character's spans, corresponding respectively in the FSA to adjacent states with either one transition or two transitions. The FSAs extracted from pairs of words in the corpora (here a Swahili translation of the Bible) are then collapsed, disambiguated, and scored in a manner reminiscent of the MDL.[13] The most robust FSAs are finally used to hypothesize stems heuristically from words not yet analysed in the corpus.[14]

[11] There is an extra layer of problem for morphology learning, due to the fact that the segmentation is not observed; the model still remains tractable using EM or any other estimation technique.

[12] This could probably be replaced by a proper MCMC estimation process, even though recursive decompositions might not be so easy to control.

[13] This score is based on the number of letters "saved" by the FSA template when generating the corresponding words.

[14] If 3 distinct words can be generated by a FSA provided a new stem is added to the production rules of one of its states, then this stem is added to the morphological template corresponding to the FSA.

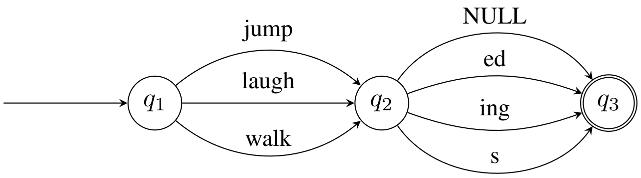


Figure 3 – The signature presented in Figure 2 now seen as a 3-states finite-state automaton

2.4.2 Learning paradigms with topic models

The work of Chan (2006) differs from most work on morphology learning in several ways: most strikingly, it does not target the identification of morpheme-like minimal units, but instead focuses on the identification of paradigms, i.e. of sets of suffixes associated with morphological properties and such that each element of such sets can be freely attached to a set of stems (this concept is in fact very close to the concept of signature introduced above). This approach is midway between supervised and unsupervised, since the input to morphology learning is a list of analysed (ie. decomposed as a stem and a suffix) words; learning however uses unsupervised learning techniques, here the Latent Dirichlet Analysis of Blei et al. (2002), in a rather peculiar way. The analogy with standard LDA is a follows: "documents" are suffixes, "words" are stems, and "topics" are paradigms. With this setting, the output of LDA is a suffix/paradigm matrix on the one hand (ideally, it should be binary), and a paradigm/stem matrix on the other hand (ideally it should also be such that each stem has a non-zero probability for exactly one paradigm). To achieve its intended goals, the author uses the output of LDA to recursively split the initial suffix set into two sub-groups (to enforce the “one suffix in one paradigm condition”), and reanalyses the two resulting datasets with LDA; this procedure is iterated until the tree leaves meet some purity constraints. A straightforward reinterpretation of this work is to view it as an attempt, given that the segmentation problem is solved, to develop the analogy expressed by Equation (1) between document indexing and morphological decomposition – here using LDA instead of PLSA.

2.5 Non-parametric Bayesian models to the rescue

2.5.1 Unsupervised segmentation

Goldwater et al. (2009) recap up a series of papers on word segmentation using Bayesian non-parametric models. Note that the task is slightly different from unsupervised morphology learning, as it aims at inferring a segmentation of an unsegmented stream of symbols into minimal units, a trend of research already explored e.g. in (Deligne and Bimbot, 1997) (see § 2.2). In the case of Goldwater and her colleagues, the main focus is the analysis of utterances addressed to children, using the data of (Brent and Cartwright, 1996). In its most simplistic form, the segmentation model is a one state semi-Markov model, boosted with two additional technicalities:

— the emission distribution has a non-finite number of possible outcomes (hence the term non-parametric); technically, it relies on Dirichlet Processes and it is crucially able to produce power law ("Zipfian") distributions over words.

— the base distribution of this Dirichlet Process is a unigram of characters (letters or phonemes);

This means that during the estimation of the model, any character string of arbitrary length can be considered for inclusion in the lexicon, with a base probability decreasing exponentially with the length. This model lends itself well to MCMC-based estimation techniques (see appendix A). In addition to the initial presentation of the model in (Goldwater et al., 2006a), the authors have developed their approach in several ways: the most effective extension (for segmenting English child utterances) is to introduce first-order dependencies between words (i.e. bigram dependencies) through the framework of Hierarchical Dirichlet Processes (Teh et al., 2006). This addresses the main problem of the unigram model (built on the Dirichlet Process), which is to undersegment the corpus.

The robustness of this method is exemplified by the very limited variability of its result regardless of the different initializations employed during sampling. Its generative formulation also provides a very precious tool to interpret the various results.

Two limitations must however be stated:

— acoustic variability is ignored since the segmentation is performed on a phonemic transcription that guarantees consistency for each occurrences of the same word type;

— the inference techniques employed use "batch" learning rather than "online" learning, the latter seeming more plausible as a cognitive model for language acquisition. Consequently, Pearl et al. (2010) explored different methods simulating limited resources, while Börschinger and Johnson (2012) proposed an online method for the word segmentation problem, refining a particle filter algorithm introduced by Börschinger and Johnson (2011).

This trend of research is further developed by Mochihashi et al. (2009), who explore a nested hierarchical Pitman-Yor (HPY) language model. In this model, an additional Pitman-Yor spelling (character-based) model is embedded in a Pitman-Yor word model similar to the bigram model introduced by Goldwater et al. (2006a) but generalized to an n-gram word model (Teh, 2006). Mochihashi et al. (2009) build, in order to make the inference step tractable, a blocked (sentence-wise) Gibbs sampler using a forward-backward procedure for generating samples from a HMM. A final technicality is the use of a Poisson distribution on word length, to mitigate the implicit length model induced by the n-gram of characters, where the probability decreases exponentially. Combining the effect of (a) a larger order, (b) a better prior on words and (c) a more effective sampling procedure, the authors report a better recall (and F-mesure) than Goldwater et al. (2006a).

The work of Neubig et al. (2010) should finally also be mentioned here, building on the HPY language model, using up to 3-grams for the spelling model and the word model, with the particularity of using a phoneme lattice instead of a phonemic transcription as an input, a strategy proven successful in mitigating the effects of a noisy input.

2.5.2 Back to morpheme-based morphology

A parallel thread, which is more directly relevant to unsupervised morphology learning, is initiated in (Goldwater et al., 2006b), where the authors manage to abstract away from the model of Goldwater et al. (2006a). Rather than seing the word generation process as primarily driven by the seating arrangements, which can be supplemented by a base distribution, the viewpoint is reversed: word generation is modeled by a unigram model, mixed with a "cache" model which memorizes the "past" decisions of the generation process (this is the seating arrangement): the cache (assuming it works in a rich-gets-richer manner) thus ensures that the output distributions will have the right shape of token distribution that is observed in language processing (e.g. power-law, a.k.a Zipfian distribution). In fact, this view applies to any probabilistic model – this is the main intuition behind so-called "adaptor models". As a first application of this idea, Goldwater et al. (2006b) revisit the model implied by equation (1), repeated here for convenience (see Equation (7) on page 6):

$$P(w = tf) = \sum_{\sigma} P(\sigma)P(t|\sigma)P(f|\sigma)$$

Adaptation here will mean to separately adapt the multinomial distribution generating the prefix and suffix and integrate the caching mechanism. Estimation is performed again using MCMC, where the analysis of each word is repeatedly sampled conditioned on all the other words analyses.

2.5.3 Adaptor Grammars

Framework In an attempt to combine the learning mechanisms presented in sections 2.5.1 and 2.5.2 in a single framework, which would be able to learn language structure at different levels (segmentation of utterances into words and segmentation of words into morphemes), Johnson et al. (2007a) introduce the Adaptor Grammars.

Adaptor Grammars are an extension to probabilistic context-free grammars (PCFGs) relaxing the assumption that each subtree of a nonterminal node is generated independently from other subtrees rooted in the same nonterminal.

More formally, if we consider a PCFG defined by the quintuple (N, W, R, S, θ), where N is a finite set of nonterminal symbols, W a finite set of terminal symbols disjoint from N, R a finite set of rules of the form $A \to \beta$, with $A \in N$ and $\beta \in (N \cup W)^*$, S a particular nonterminal start symbol, and θ defining probabilities for production rules associated to each nonterminal, the distributions G_A over nonterminal [15] symbols A are defined through the following recursion:

$$G_A = \sum_{A \to B_1 \dots B_n} \theta_{A \to B_1 \dots B_n} \ \mathrm{TD}_A(G_{B_1} \dots G_{B_n}) \tag{3}$$

with the tree distribution TD_A defined by

$$\mathrm{TD}_A(G_{B_1} \dots G_{B_n}) \left(\begin{array}{ccc} & A & \\ t_1 & \dots & t_n \end{array} \right) = \prod_{i=1}^{n} G_{B_i}(t_i) \tag{4}$$

[15] For terminal symbols A, G_A is the distribution putting all of its mass on the single node labelled A.

To define an Adaptor Grammar from this PCFG, we consider the "adaptors" C_A for $A \in N$, with each C_A defined as a function from the distribution G_A to a distribution over distributions having the same support as G_A. The recursion defining the new distribution H_A over nonterminal symbol A is given by:

$$H_A \sim C_A(G_A) \tag{5}$$

$$G_A = \sum_{A \to B_1 \dots B_n} \theta_{A \to B_1 \dots B_n} \ \mathrm{TD}_A(H_{B_1} \dots H_{B_n}) \tag{6}$$

For example, the adaptor C_A can be the function associating a distribution G_A to the Dirichlet Process $\mathrm{DP}(\alpha_A, G_A)$ (see Appendix A.1.3) with concentration parameter α_A and base distribution G_A. It is also possible to define C_A as the identity [16] function for certain "non-adapted" nonterminals. If the set of adapted nonterminals is denoted by M, we obtain the Adaptor Grammar associating the distributions H_A over each nonterminal A defined by:

$$\begin{aligned} H_A &\sim \mathrm{DP}(\alpha_A, G_A) && \text{if } A \in M \\ H_A &= G_A && \text{if } A \notin M \\ G_A &= \sum_{A \to B_1 \dots B_n} \theta_{A \to B_1 \dots B_n} \ \mathrm{TD}_A(H_{B_1} \dots H_{B_n}) \end{aligned}$$

Using these adaptors in the recursion, it is possible to allow for a symbol's expansion to depend on the way it has been rewritten in the past. Informally, Adaptor Grammars are able to "cache" entire subtrees expanding nonterminals and provide a choice to rewrite each new nonterminal either as a regular PCFG expansion or as a previously seen expansion. In this respect, Adaptor Grammars can be seen as a nonparametric extension of PCFGs. Moreover, using adaptors based on the Dirichlet process or the Pitman-Yor process, one can build models capturing certain power-law distributions observed in natural language.

Inference The training data in this unsupervised context consists only in terminal strings (*yields* of trees rooted in the start symbol S). In order to sample (posterior distributions over) analyses produced by a particular Adaptor Grammar based on Pitman-Yor processes, Johnson et al. (2007a) devise a method relying on a Markov chain Monte Carlo algorithm (see Appendix A.2.1) together with a PCFG approximation [17] of the Adaptor Grammar. The idea for this approximation is, for each analyzed string, to add to the rules of the "base" PCFG all the production rules corresponding to the yields of the adapted nonterminals in the Adaptor Grammar, given all the (analysed) strings in the data set except the currently analysed string. One can sample analyses from this PCFG using the algorithm described in (Johnson et al., 2007b).

The inference procedure follows the following main steps:

[16] More precisely, as a function mapping G_A to the distribution placing all its mass on G_A in the space of distributions.

[17] After relaxing the independence assumption made in PCFGs, there is apparently no efficient direct sampling procedure from $P(u_i|s_i, \boldsymbol{u}_{-i})$, with u_i the analysis of the ith string s_i in the data, and $\boldsymbol{u}_{-i}$ the vector of all the analyses except analysis u_i, required to perform a MCMC procedure.

1. Initialize with a random tree generated by the grammar for each string,
2. Randomly select a string and sample a parse from the PCFG approximation,
3. Update the parse for this string if the (Metropolos-Hastings) procedure accepts the proposed analysis,[18]
4. Go back to step 2 until convergence. At convergence, the analyses are samples of the posterior distribution over analyses under the Adaptor Grammar, and can be used to compute the production probabilities θ.

Expressivity The Adaptor Grammar framework's main strength lies in its flexible and powerful expressivity. It is possible to replicate under this framework equivalent or similar non-parametric models already discussed in the context of word segmentation (Goldwater, 2006; Goldwater et al., 2006a, 2009) or morphology learning (Goldwater, 2006; Goldwater et al., 2006b). For example, the unigram model for word segmentation presented in (Goldwater et al., 2006a) corresponds to the following production rules in the Adaptor Grammar (the adapted nonterminal has been underlined):

$$\begin{array}{l} \text{Sentence} \rightarrow \underline{\text{Word}} \\ \text{Sentence} \rightarrow \underline{\text{Word}}\ \text{Sentence} \\ \underline{\text{Word}} \rightarrow \text{Phonemes} \\ \text{Phonemes} \rightarrow \text{Phoneme} \\ \text{Phonemes} \rightarrow \text{Phoneme Phonemes} \end{array}$$

But more importantly, it is also possible to infer in a single procedure over structures that are mutually dependent, for example word boundaries and word-initial syllable collocations. In other words, it is possible to learn simultaneously something about the structure of an utterance and the structure of the words composing it. This requires to specify more than one adapted nonterminal in the grammar which turns out to be equivalent to implementing a hierarchical Dirichlet process (HDP). Hence, integrating a learning procedure for morphology into the grammar learning word segmentation just presented would consist in adding for example rules of the form:

$$\begin{array}{l} \underline{\text{Word}} \rightarrow \underline{\text{Stem}} \\ \underline{\text{Word}} \rightarrow \underline{\text{Stem}}\ \underline{\text{Suffix}} \\ \underline{\text{Stem}} \rightarrow \text{Phonemes} \\ \underline{\text{Suffix}} \rightarrow \text{Phonemes} \end{array}$$

while removing the "$\underline{\text{Word}} \rightarrow \text{Phonemes}$" production.

It should be noted however that the expressivity of this framework presents some limits, since the number of adaptors is required to be fixed in advance and corresponds to the number of nonterminals. Goldwater's bigram model (Goldwater et al., 2009) evoked in Section 2.5.1 for example, associates one Dirichlet Process per word type, and the number of these types is not known in advance. Johnson (2008b) shows that introducing a word collocation adapted nonterminal is able to nonetheless capture inter-word dependencies and achieves similar performance to Goldwater et al.'s bigram model.

[18] This step corrects the probability approximation made using the PCFG "snapshot" of the Adaptor Grammar.

Experiments Experiments on the english corpus of child-directed speech already mentioned in 2.5.1 are performed in (Johnson, 2008b; Johnson and Goldwater, 2009; Johnson et al., 2014) with successive improvements relying on increasingly complex grammars, taking into account phonotactic constraints, and different levels of collocations, together with refined initialization, advanced sampling techniques,[19] and modeling of function words.

Using a different set of data, Johnson (2008a) looks at unsupervised morphology learning for a Bantu language, Sesotho. This is mostly an experimental work exploring various ways to express morpho-phonological knowledge into the formalism of Adaptor Grammars; the task is to segment in words children productions. One interesting outcome of this study is to show the effectiveness of having an explicit hierarchical model of word internal structure for Sesotho, which is not entirely surprising given the complexity of Sesotho's morphology, and the applicability of the Adaptor Grammar framework to a language morphologically very far from English. In comparison, modeling the context does not seem as important as it is for English (Goldwater et al., 2009). We reproduce in Figure 4 a structured output from (Johnson, 2008a), capturing word collocations and morphemes in a character [20] sequence in Sesotho.

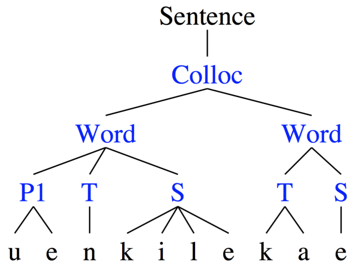


Figure 4 – A parse tree from (Johnson, 2008a) generated by an adaptor grammar specifying expansion rules for word collocations and morphemes in a character sequence in Sesotho.

2.5.4 Extensions and other works

More recent work has been building on the techniques presented in the preceding sections. O'Donnell et al. (2009), elaborating on the idea of a heterogeneous lexicon, introduce Fragment Grammars, a generalization of Adaptor Grammars in which fragments of subtrees can be adapted – and not only entire subtrees yielding terminal strings –, that is to say, the distribution of a subtree prefix can be learnt in this framework. It is not clear however to which degree this extension compares to state-of-the-art results on standard tasks. Botha and Blunsom (2013) propose another extension to the Adaptor

[19] Resampling of the table labels within the Gibbs sampling procedure, sampling of the adaptor's hyperparameters, and integrating out the production rules' probabilities.

[20] The Sesotho orthography, very close to phonemic, has not been phonemicized here.

Grammar formalism using a probabilistic and adapted version of Simple Range Concatenating Grammars (SRCGs) attempting to capture non-concatenative phenomena in morphology and obtaining improvements in a task of morphological segmentation for semitic languages (in this case Arabic and Hebrew).

We should also mention the work of Cohen et al. (2010) who devise a variational inference algorithm which provides an alternative to MCMC methods used during inference for Adaptor Grammars. A recent work (Zhai et al., 2014) combines both methods in an online and hybrid fashion, significantly improving the inference's speed for Adaptor Grammars. Synnaeve et al. (2014), lastly, take advantage of non-linguistic context to improve word segmentation using an Adaptor Grammar. This non-linguistic context (activity at stake, visual cues) is approximated as a topic obtained via the training of a topic model (Blei et al., 2002). The most probable topic of each utterance is then added as a prefix and the grammar is modified to make use of them. This proves to be helpful for the task of word segmentation.

2.6 Learning in word-based morphology

The work of Dreyer and Eisner (2011) is the culmination of a series of papers (Dreyer et al., 2008; Dreyer and Eisner, 2009) aimed at learning *word-based models of morphology* (Aronoff, 1976; Blevins, 2006). Under this view of morphology, morphological processes cannot be reduced to the concatenation of segmental strings to a stem; instead, morphology should attempt to model the relations between forms within *paradigms* – a notion that should therefore be given a first class status in this theory.

From a computational perspective, segmentation is no longer the main target; instead, the model should be able to (i) cluster related forms within paradigms; (ii) learn the mapping between forms and slots in the paradigm; (iii) predict the realization of slots that are not observed in the corpus, all this in an (almost) unsupervised fashion.[21] In a nutshell, this work relies on two main components: (a) a finite-state probabilistic model for morphologically related forms, which should capture the surface similarity (and systematic alternations) between forms within a paradigm; (b) a non-parametric Bayesian model which takes care of the statistical regularities of the distribution of types, inflections, and forms

Regarding (a), more fully described in (Dreyer and Eisner, 2009), the main contribution is a probabilistic model for *sets of strings* composing a paradigm. Technically, the model is a Markov Random field, where the probability of a set of strings Π is defined as:

$$P(\Pi = \{w_1, \ldots, w_K\}) = \frac{1}{Z} \prod_{i,j} F_{ij}(w_i, w_j), \tag{7}$$

where each factor F_{ij} is a weighted transducer which computes a parameterized similarity score between w_i and w_j. This model is trained using (loopy) belief propagation: recall that when the random variables in the MRF are discrete, the BP algorithm

[21] In the work of Dreyer and Eisner (2011), the supervision consists mostly of an abstract description of the paradigm's cells and of a handful of exemplar paradigms.

sends messages between factors and nodes that take the form of probability tables (indexed by the variable values); in this more complex case where variables range over string domains, the messages are distributions over languages, represented as weighted Finite-State Automata (FSA). Assuming that factors are computed with Finite-State Transducers, the authors then show how to implement message passing using basic FSA / FST operations (composition, intersection, projection). Experiments show how to use this framework to predict likely (unknown) string values for partially observed paradigms (for German verbs, using a restricted definition of paradigms).

Regarding (b), the main idea is to model the occurrences of word forms (given their POS) as the combined result of the uneven distribution of lexemes in the one hand; of the uneven distribution of inflections in the other hand. Both are modeled again using the ubiquitous Dirichlet Processes, estimated using Gibbs sampling (see Appendix A). This work contains many subtle technicalities, regarding for instance the parameterization of the inflection model, or the approximations needed to control the convergence of messages during LPB. Experiments using German conjugation demonstrate the potential of the method to generate unseen forms for a verb based on handful of realizations.

From a technical point of view, this line of work is more recently developed in (Cotterell et al., 2015) and (Peng et al., 2015), where alternative approximate inference techniques (eg. dual decomposition) are considered for solving string similarity problems.

2.7 Miscellaneous

It would be out of the scope to cover the entirery of the literature on unsupervised morphology learning (the proceedings of the Morpho challenge evaluation campaigns[22]), but the work of De Pauw and Wagacha (2007) deserves some discussion, since it focuses on a Bantu language (Gĩkũyũ) spoken in Kenya. The authors describe a two step strategy to identify prefix / stem combinations. In a first step, the authors train a probabilistic exponential model computing the conditional probability of a word w, given a vector of orthographical features $\phi(w)$, as:

$$P(w|\phi(w);\theta) = \frac{exp(\theta^T F(\phi(w), w))}{Z(\theta, \phi(w))},$$

where the parameters θ are trained by maximizing the likelihood. Note that this should yield very poor parameter estimates, since every class is only viewed once in the training data. It is then possible to compute the closest neighbours w' to any word w by ranking lexical entries according to $P(w|\phi(w');\theta)$. Once this list of k-closest neighbours is computed, the authors process it to compute frequent transformations between ressembling words. The more frequent graphical transformations are retained, yielding a list of candidate prefixes in the language, which is then compared to a reference listing.

[22] http://research.ics.aalto.fi/events/morphochallenge/

2.8 Summary

Over the past ten years, a very strong current among the researchers involved with the unsupervised learning of morphology has been leading to the substitution of heuristics-based methods with more principled Bayesian methods. Mostly directed towards the learning of concatenative or synthetic morphology, which exhibits a relative identity between form and meaning, these methods achieve impressive results – given the absence (or limited quantity) of supervision – especially in the context of a segmentation task, which has attracted most of the interest of the community so far.

Some work has already been initiated to extend these models to templatic, non-concatenative, languages but remain marginal. Moreover, tonal languages, like the ones we are focussing on in the BULB project, have been given much less attention, and there is, to this day and to our knowledge, no theoretical framework incorporating tones for the learning of morphology. New work could also address the decomposition of word forms taking advantage of distributional syntactic or semantic cues.

3 Automatic word alignment

In this section, we study another relevant area of linguistic processing where unsupervised learning techniques have proven to be very effective: the problem of aligning parallel sentences, comprised of a source sentence and its target translation, at the level of words. Such techniques have been mostly developed and used in the context of statistical machine translation, and are more fully documented in recent textbooks or reviews (Koehn, 2010; Tiedemann, 2011; Wu, 2012).

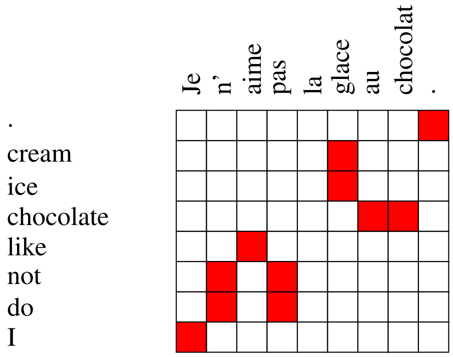
In this review, we mostly focus on the family of generative word alignments initially proposed in (Brown et al., 1990), which are a requirement for the third part of this article. We also mention in passing alternative, more recent, approaches that now are part of the state-of-the-art in word alignment.

This section is organised as follows: we first define the notion of alignment in § 3.1, before moving one with the simplest models which are essentially variants of HMMs (§ 3.2 and 3.3); § 3.4 then gives an overview of the more complex IBM Model 3 and of its derivative. We conclude with two brief sections discussing symmetrization issues, and pointing to subsequent further readings (§ 3.6).

3.1 Building word alignments

A parallel corpus is a made of sentence pairs which are mutual translations: the source sentence $\mathbf{f}$ is a sequence of J words $\mathbf{f} = f_1, \ldots, f_j, \ldots, f_J$ and the target sentence $\mathbf{e}$ is a sequence of I words $\mathbf{e} = e_1, \ldots, e_i, \ldots e_I$. Word alignments between $\mathbf{f}$ and $\mathbf{e}$ relate words which are mutual translations. A first possible representation of an alignment between two sentences is a binary matrix $\boldsymbol{A} = (a_{i,j})$, where each cell $a_{i,j}$ indicates whether word f_j is aligned with the word e_i, as in the example in Figure 5.

Under the matrix representation of word alignments, there are $2^{I \times J}$ possible alignments, as many as the number of possible values for $\boldsymbol{A}$. Rather than directly modeling these matrices, which yields intractable combinatorial problems, simpler representa-



	Je	n'	aime	pas	la	glace	au	chocolat	.
.									🟥
cream						🟥			
ice						🟥			
chocolate							🟥	🟥	
like			🟥						
not		🟥		🟥					
do		🟥		🟥					
I	🟥								

Figure 5 – Example of an alignment matrix between an English sentence and a French sentence. The non-zero cells in the matrix are represented by red squares. The set of links associated with this matrix is {(1,1), (2, 2), (2, 4), (3, 2), (3, 4), (4,3), (5, 7), (5, 8), (6, 6), (7, 6), (8, 9)}

tions are usually considered. An effective way to restrict the number of alignments is to only consider *applications* from $[0 : I]$ into $[0 : J]$, of which there are only I^J. Under this simplifying assumption, each word of one of the sentences is labeled with the position index of the corresponding word in the other sentence. The model is no longer symmetric.

Building such an alignment then amounts to finding the label sequence $\mathbf{a} = a_1, \ldots, a_j, \ldots a_J$ associated with $\mathbf{f}$, where, for a given sentence pair, each label a_j belongs to the set of positions in the target sentence: $a_j \in \mathcal{A} = 0, 1, \ldots, I$. a_j represents the index in $\mathbf{e}$ with which f_j is aligned: the word f_j is therefore aligned with the word e_{a_j}. When a word cannot be aligned, we use the conventional label $a_j = 0$. Licensing null alignments thus introduces an additional empty word $e_0 = \texttt{null}$ in the target sequence, which is then composed of $I + 1$ words.

Word alignment models were introduced in (Brown et al., 1990). In this pioneering work, the word alignments represent hidden variables of the word-based translation process. In the following presentation of word alignments, aspects related to translation are deliberately left aside.[23] The insightful presentation of the most common alignment models given in (Och and Ney, 2003) will be our main source of inspiration. This presentation covers the IBM models of (Brown et al., 1990) as well as the hidden Markov model (HMM) proposed in (Vogel et al., 1996) and several additional heuristic models.

Modeling alignment as a sequence labeling problem can be presented by decom-

[23] The presentation of these models in a word-based translation framework would involve numerous details which are omitted here. It is nowadays accepted that word-based systems underperform phrase-based systems, and are no longer used.

posing the joint probability of a sequence of observations (the source sequence $\mathbf{f}$) and of the associated label sequences $\mathbf{a}$ (alignments). This probability is conditioned here by the target phrase $\mathbf{e}$ which is assumed to be known and which restricts the realization space of the alignments. Equation (8) will be our starting point:

$$\mathrm{P}(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \mathrm{P}(\mathbf{a}|\mathbf{e})\, \mathrm{P}(\mathbf{f}|\mathbf{a}, \mathbf{e}) \tag{8}$$

This equation introduces two terms on the right hand side, corresponding respectively to the conditional probability of the label sequence, $\mathrm{P}(\mathbf{a}|\mathbf{e})$, and that of the observation sequence given the labels $\mathrm{P}(\mathbf{f}|\mathbf{a}, \mathbf{e})$. The latter term is usually denoted as the *translation probability*, and the former the *distortion probability*.

Starting from (8), Brown et al. (1990) study four generative models of increasing complexity denoted by the nicknames IBM model 1–IBM model 4.[24] Each of this model corresponds to a set of symplifying assumptions regarding the dependencies between the random variables of Equation (8), yielding models of increasing complexities.

3.2 IBM model 1

IBM model 1 is based on the following assumptions:

- each alignment link a_j is independent from the other alignments and is uniformly distributed, implying that:

$$\mathrm{P}(\mathbf{a}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j|\mathbf{e}) = \prod_{j=1}^{J} \frac{1}{I+1} = \frac{1}{(I+1)^J}$$

- given a_j, each source word $\mathbf{f}$ only depends on the target word e_{a_j} it aligns with:

$$\mathrm{P}(\mathbf{f}|\mathbf{a}, \mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(f_j|\mathbf{a}, \mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(f_j|e_{a_j})$$

Under these conditional independence assumptions, equation (8) can be simplified as follows:

$$\mathrm{P}(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j,\, f_j|e_{a_j}) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \mathrm{P}(f_j|e_{a_j}) \tag{9}$$

This model is only parameterized by the set of conditional distributions modeling the translation equivalents of each target word e; this model is therefore often described as a *lexical translation model*. Accordingly, the parameter vector $\boldsymbol{\theta}$ of the IBM model 1 model is written as $\boldsymbol{\theta} = \{\mathrm{P}(f|e), \forall (f, e) \in \mathcal{V}_f \times \mathcal{V}_e\}$. In practice, the vocabulary $\mathcal{V}_f$ (respectively $\mathcal{V}_e$) only contains the words that are seen in the source (respectively target) side of the parallel corpus.

[24] This series was later completed and actually runs till model 6 (Och and Ney, 2003)

Knowing the parameters $\boldsymbol{\theta}$, it is easier to infer the most probable alignment for any pair of sentences by applying the maximum *a posteriori* rule:

$$\mathbf{a}^* = \operatorname*{argmax}_{\mathbf{a}} \mathrm{P}(\mathbf{a}|\mathbf{f},\mathbf{e}) = \operatorname*{argmax}_{\mathbf{a}} \mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e}) \tag{10}$$

The search for the best alignment $\mathbf{a}^*$ is particularly efficient as the evaluation of each alignment link a_j is independent of the other alignments. The most probable sequence $\mathbf{a}^*$ is thus the one that maximizes the joint probability $\mathrm{P}(\mathbf{a},\mathbf{f}|\mathbf{e})$. From Equation (9), this joint probability is a product of terms in the internal $[0:1]$. This product is maximum when each term is itself a maximum:

$$\forall j, a_j^* = \operatorname*{argmax}_{a_j \in \mathcal{A}} \mathrm{P}(f_j|e_{a_j}) \tag{11}$$

Estimation can be straightforwardly performed in a supervised fashion, i.e. given a set of parallel sentences and their word alignments. Unfortunately, this type of data is rarely available in sufficient quantity. It is however possible to train the model in a unsupervised way from a sentence aligned bilingual corpus. In this setting, alignments correspond to hidden variables and parameter estimation has to resort to numerical procedures such as the *Expectation-Maximization* (EM) algorithm of (Dempster et al., 1977), so as to maximize the *log-likelihood* of the parameters on the learning corpus. The specific assumptions of IBM model 1, in which each word is aligned independently from its context, make this optimization problem tractable. In addition, Brown et al. (1993) show that the *log-likelihood* is concave [25] which guarantees the convergence of the algorithm toward an overall optimum. Implementing the EM algorithm for the IBM model 1 is performed as follows:

— *Initialization:* randomly or uniformly set initial values for $\boldsymbol{\theta}$;

— *Until convergence or for a predefined number of iterations*:

   — *E-step:* knowing $\boldsymbol{\theta}$, the model is used to compute the posterior distribution of each alignment link:

$$\mathrm{P}(a_j|f_j,\mathbf{e}) = \frac{\mathrm{P}(a_j, f_j|\mathbf{e})}{\mathrm{P}(f_j|\mathbf{e})} = \frac{\mathrm{P}(f_j|e_{a_j})}{\sum_{i=0}^{I} \mathrm{P}(f_j|e_i)} \tag{12}$$

Knowing the posterior probabilities, the expectation of the number of occurrences of an alignment between f and e in the parallel corpus is computed by summing the posteriors over all sentence pairs $(\mathbf{f},\mathbf{e})$:

$$\#(f|e) = \sum_{(\mathbf{f},\mathbf{e})} \left[\mathrm{P}(a|\mathbf{f},\mathbf{e}) \left(\sum_{j=1}^{J} \delta(f, f_j) \right) \left(\sum_{i=0}^{I} \delta(e, e_i) \right) \right] \tag{13}$$

In Equation (13), a denotes the alignment link between f and e, and $\delta(,)$ is the Kronecker function whose values is 1 if both arguments are equal, and 0 otherwise.

[25] But not strictly concave, which implies that the optimum is not necessarily unique (Toutanova and Galley, 2011). A workaround this problem, which restores concavity is discussed in (Simion et al., 2015).

— *M-step:* Once these expectations have been computed, parameters $\boldsymbol{\theta}$ are estimated as follows :

$$\mathrm{P}(f|e) = \frac{\#(f|e)}{\sum_{f' \in \mathcal{V}_f} \#(f'|e)} \tag{14}$$

The IBM model 1 is a very simple alignment model, which only uses the cooccurrence information between source and a target words in parallel sentences to estimate the translation model according to equation (13). Although rudimentary, this model obtains good results when very large amounts of data are available (Brants et al., 2007).

3.3 Computing alignments with hidden Markov models

One obvious weakness of model 1 is the hypothesis that all the alignment links a_j are equally likely: this is especially false for related language, where the word order is globally (if not locally) preserved. For example, with the French/English language pair, a source word at the beginning of a sentence is rarely aligned with a target word at the end of the sentence. To take this information into account, the IBM model 2 of Brown et al. (1990) introduces a dependency between the value of a_j and the absolute position j in the sentence, expressed by the term $\mathrm{P}(a_j|j, J, I)$. This dependency is used to favor certain alignments over others. A very efficient and effective implementation of model 2, based on a reparametrization of the distortion model which enables a straightforward computation of link posteriors, is introduced in (Dyer et al., 2013).

This model misses another important point, though: the tendency of alignment to respect some kind of *monotonicity*. This means that if the source word at position $j-1$ is aligned with the target word at position a_{j-1}, it is then highly likely that the source word at position j will be aligned with the target word at position $a_j = a_{j-1} + 1$, or at least, with a word nearby. In other words, it seems relevant to model the "jump" between two consecutive alignments based on the difference $\mid a_j - a_{j-1} \mid$. As an illustration, take the case of a sequence composed of a determiner followed by a noun: in such situation, the alignment a_j of the noun is influenced by the alignment a_{j-1} of its preceding determiner. The monotonic character of alignments is better modeled with a first order hidden Markov model (HMM), which introduces a dependency between the alignment links of two consecutive words a_{j-1} and a_j (Vogel et al., 1996). . The main assumptions of the model are modified as follows (to be compared to IBM Model 1).

— each "label" a_j depends on the value of the previous label a_{j-1}. The probability of a label sequence is therefore written as [26]:

$$\mathrm{P}(\mathbf{a}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j|a_{j-1}, \mathbf{e})$$

— each word f_j in $\mathbf{f}$ only depends on a_j, or rather, on the target word in position a_j.

[26] Assuming a conventional value for a_0.

Under the hypothesis of first order Markovian dependencies, the joint probability of the source sequence and the associated label sequence is simplified as follows:

$$\mathrm{P}(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \prod_{j=1}^{J} \mathrm{P}(a_j|a_{j-1}, \mathbf{e})\, \mathrm{P}(f_j|e_{a_j}) \tag{15}$$

where, as for the IBM model 1, $\mathrm{P}(f_j|e_{a_j})$ denotes the probability of translating word e_{a_j} by f_j. In order to parameterize the transition probabilities $\mathrm{P}(a_j|a_{j-1}, \mathbf{e})$, Vogel et al. (1996) make an additional assumption, stipulating that this transition probability only depends on the size of the jump between a_{j-1} and a_j:

$$\mathrm{P}(a_j|a_{j-1}, \mathbf{e}) = \frac{s(a_j - a_{j-1})}{\sum_{a \in \mathcal{A}} s(a - a_{j-1})} \tag{16}$$

where $\{s(j - j')\}$ is a set of positive parameters. The set of parameters $\boldsymbol{\theta}$ of the model therefore comprises the same parameters as the IBM model 1, complemented with the parameters related to the transition probability. Knowing the model parameters $\boldsymbol{\theta}$, the most probable alignment for a given sentence pair is determined using equation (10). The introduction of dependencies between two successive labels makes the problem more complex than the equivalent problem for the IBM model 1. It can however be solved efficiently using the Viterbi algorithm.

Like for the IBM model 1, parameter estimation from unlabeled data can be performed using the EM algorithm. It involves the following steps:

- *Initialization:* randomly or uniformly set the initial values for $\boldsymbol{\theta}$.
- *Until convergence or for a predefined number of iterations*:
  - *E-step:* the dependencies between labels makes the computation of link posteriors more challenging, since it can no longer be done on a per word basis. Yet, using standard algorithms for HMMs such as the Forward-Backward algorithm, link posteriors $\mathrm{P}(a_j|\mathbf{f}, \mathbf{e})$ are readily derived. Likewise, it is also possible to compute jump posteriors $\mathrm{P}(i, i'|\mathbf{f}, \mathbf{e})$ where i and i' are the labels of two consecutive target words. Once these posterior probabilities are known, it is possible to compute the expectations of word association counts $\#(f|e)$ and of jump counts $\#(i|i')$ by summing over all sentences and alignment links.
  - *M-step:* the parameters $\boldsymbol{\theta}$ of the model are directly re-estimated by normalizing the expectations of the counts as in equation (14).

An important difference between the HMM model and the IBM1 model is that the *log-likelihood* is no longer concave and it presents numerous local extrema, removing any convergence guarantee toward a global maximum. In numerous application frameworks, this problem is not considered harmful as long as the initial values of the parameters are properly set. The most common choice consists of initializing the lexical association parameters $\mathrm{P}(e|f)$ with values of a previously estimated IBM1 model.

3.4 Modeling fertility, IBM model 3 and beyond

When aligning a sequence $\mathbf{f}$, each word f_j is linked to a unique target word e_{a_j}. This hypothesis is obviously limiting, since it often occurs that the same concept is expressed by word groups of different lengths in the source and target languages. This problem is illustrated in Figure 6, which displays two extracts of parallel sentences. In the first example, the French compound "pommes de terre" is linked with a single English word *potatoes*. When aligning French words with positions in the English sentence, the alignment model can find a consistent solution for aligning the three source words *pomme*, *de* and *terre* with the same target word *potatoes*. However, depending on the training data, another solution can emerge which aligns *pomme* with *potatoes* and the two other source words with `null`.

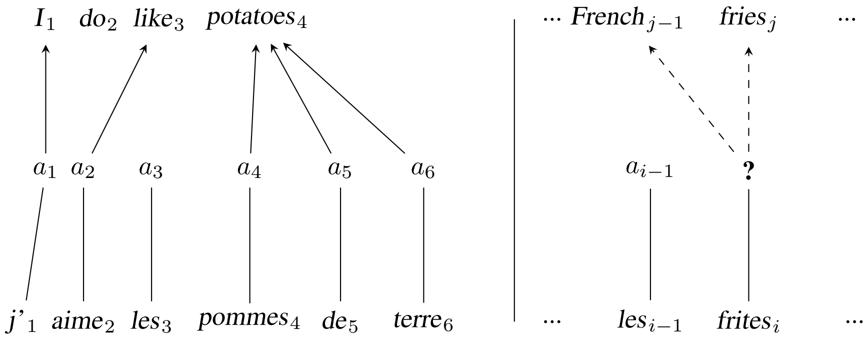

I_1 do_2 $like_3$ $potatoes_4$
a_1 a_2 a_3 a_4 a_5 a_6
j'_1 $aime_2$ les_3 $pommes_4$ de_5 $terre_6$
... $French_{j-1}$ $fries_j$ ...
a_{i-1} ?
... les_{i-1} $frites_i$ ...

Figure 6 – Fertility, a problem for sequence labeling approaches

A manner of guiding the alignment towards the "good" solution consists of introducing the notion of *fertility*: each target word is associated with a distribution $n(\phi|e)$ modeling the number of associated source words. This new distribution captures the tendency of certain target words to receive more (or less) than one single alignment links. For example, the probability distribution in Table 1 reflects the fact that *potatoes* prefers to align either with a single word or with three words.

ϕ	$n(\phi\|\text{potatoes})$
0	0
1	0.42
2	0.1
3	0.48
4	0

Table 1 – Fertility distribution for the word *potatoes*

Fertility can also model the preference of certain target words to remain unaligned. This is, for example, the case of an auxiliary verb such as *do* in the example of Figure 6. Such phenomena will be simply modeled by assigning a non-zero probability to the event corresponding to leaving *do* unaligned.

Another interesting case in this example is the source word *les*, which must remain unaligned, that is, aligned with the `null` word. This phenomenon can be accounted for in the IBM1 and HMM models; it is however possible to refine the prediction of null alignments using the fertility model. Based on the intuition that, within a given sentence, the number of words aligned with `null` mainly depends on the sentence length, Brown et al. (1990) propose to use a particular distribution for the fertility of the `null` word, $p(\phi_0)$, which depends on the sentence length and of a parameter p_0 quantifying the prior probability of a null alignment.

To sum up, in the IBM model 3, the joint probability $\mathrm{P}(\mathbf{a}, \mathbf{f}|\mathbf{e})$ includes two new terms with respect to model 2: the fertility of the `null` word $p(\phi_0)$ and, for each target word, its fertility $n(\ |.)$ Finally, Brown et al. (1990) also introduce the notion of *distortion*, which is used to model jumps.

Model 3 therefore corresponds to a better modeling of the interactions between the two languages, at the price of an increased algorithmic complexity. This is because the model includes a much larger number of parameters than the IBM model 1 and HMM models, since it includes, for each target word, the corresponding fertility parameters.[27] Furthermore, due to the structure of dependencies between random variables, it is no longer possible to solve the inference problem (10) exactly; nor is it possible to compute the expectations needed to perform EM training.

In fact, in this new model, the search for an optimal alignment is a NP-hard problem (Udupa and Maji, 2006), that can only be solved heuristically. Similarly, failing to compute the sums over all the possible alignments that are necessary for the link posteriors, learning algorithms most commonly used sampling techniques to approximate the quantities required for the EM algorithm.

IBM models 4, 5, and 6 bring additional refinements to the model 3 that we will only briefly summarize here. Interested readers can find more information in the following papers (Brown et al., 1990; Och and Ney, 2003). IBM model 4 improves the model 3 by introducing an enhanced distortion model which takes the relative positions of the words into accounts, and also includes first order Markovian dependencies. In addition, words classes are built in the source and target languages and are used to improve the lexical association probabilities. These classes, can be computed in an unsupervised manner, from monolingual learning data (Brown et al., 1992). Model 5 corrects a theoretical problem posed by the models 3 and 4, which are said to be *deficient*. This is because a part of the probability mass is attributed to alignments which are impossible, due to fertility constraints. In fact, the sum over target words of fertility values cannot exceed the number of target words; yet, such configurations have a non zero probability. Finally, the model 6 model is a combination of the IBM Model 4 and HMM models introduced in (Och and Ney, 2003) in an attempt to combine the advantages of both models. The increase in complexity incurred by these two latter models is usually not compensated by improvements of the resulting alignments, which explains that these models are rarely used in practice.

[27] This represents $|\mathcal{V}_e| \times (F_{max})$ additional parameters, where F_{max} is the maximum value for the fertility.

3.5 Symmetrization

The examples in Figure 6 illustrates some of the limitations of viewing alignment as a mere sequence labeling problem. In the first example, three words must be aligned with a single word, which is only possible in one direction: when aligning English with French, rather than French with English, it becomes impossible to come up with a satisfactory solution. The same happens in the second example, where a single French word in "frites" is to be aligned with a pair of English words *French fries*. Here, the alignment of French with English is problematic, whereas the opposite direction poses no particular problem.

As it turns out, the two alignment directions provide complementary information, which can be merged into a unified representation using *symmetrization heuristics*. Using these heuristics, it is then possible to reconstruct symmetric alignments represented as a $J \times I$ boolean alignment matrix.

A first simple heuristic considers the *union* of all alignment links contained either in the source/target or in the target/source generative alignments. The resulting matrix may contain unsure links, which are only proposed in one alignment direction. A more conservative approach selects only links in the *intersection* of the two input alignments. In this case, the number of alignments obtained is much lower, and only contains reliable links which have been proposed in both alignment directions. Various additional heuristics are discussed in (Och and Ney, 2003; Koehn et al., 2003), aimed at completing the *intersection* alignments with some, but not all, links from the *union*.

The various symmetrization heuristics aim to reconcile *ex post* two asymmetric alignments, since symmetry of alignment seems to be a quite desirable property. Several more principled attempts have been made to enforce symmetry more directly during training. This is notably the case of (Liang et al., 2006), where the authors formulate a new training criterium, which combines the log-likelihood of two asymmetrical models plus a additional term which mimicks a soft intersection. The resulting optimization problem being untractable, the author resort to an approximate optimization algorithm, reminiscent of EM, with however a non-conventional way to compute agreeing link posteriors. More recently, this line of research is further developed in (Graça et al., 2010), where two asymmetrical HMM models are forced to agree *on average*: the trick is to replace the actual posterior link distributions in the HMM with the nearest distribution satisfying symmetry constraints, which is then plugged into the reestimation formula. Finally, the approach of DeNero and Macherey (2011) also targets symmetrical alignments, using alternative approximate inference techniques, here dual decomposition.

3.6 Summary

Word alignment models in statistical machine translation are used to compute two (one for each direction) asymmetrical alignments of parallel sentences, which are then heuristically symmetrized. An open-source implementation of the generative, unsupervised, models of Section 3.1 is available in the `Giza++` toolkit (Och and Ney, 2003).[28]

Among the alignment models presented above, the IBM4 model seems to realize the best compromise between training time and alignment quality.[29] Training such model cannot be done from scratch, and typically involves to learn a cascade of models of increasing complexity, where simpler models are used to initialize the more complex ones: IBM model 1, then HMM, model 3, model 4 and so forth (even though the standard practice is to stop after model 4).

The performance of `Giza++`, from the point of view of the quality of the predicted alignments are not entirely satisfactory. Reducing word alignment errors is still the subject of much active research, even though better alignments do not necessary yield better translations (Fraser and Marcu, 2007). For instance, some works explore the usage of discriminative models such as conditional random fields (CRFs) (Taskar et al., 2005; Ayan and Dorr, 2006; Blunsom and Cohn, 2006; Niehues and Vogel, 2008; Tomeh et al., 2014). Discriminative models (Moore, 2005) are more expressive and enable to combine multiple information sources to assess the validity of an alignment link. In particular, they can easily integrate linguistic features such as syntactic or morpho-syntactic analysis, or use surface similarities between source and target words. Training such models however requires *supervision* data, in the form of manually aligned sentence pairs. Such annotations are difficult and very costly to produce, and very few hand aligned corpora are available. This may explain why, despite their improved performance, discriminative models have not yet replaced the IBM alignment models.

4 Joint Models for segmentation and alignment

Morphological divergences between languages are, as mentioned above, a major issues for word alignments algorithms, which assume similar concepts of words on both sides of the alignment, hence making the identification of one-to-one correspondance a reasonable goal. When this hypothesis is violated, which happens for instance when one attempt to align a fusional language such as English with a concatenative language (say Finnish), alignment performance decreases significantly. This is because (a) one-to-many alignments from the concatenative to the fusional language tend to leave too many words unaligned (on the fusional side); (b) rich morphological variations multiply the number of word forms, which badly hurts the robustness of the statistical procedures implied in the alignment process.

In this final section, we survey works aimed at combining ideas from the segmentation and from the alignment literature to mitigate these issues.

Similar problems occur, and have yielded similar solutions, in areas where one side of the alignment needs to be segmented, either because the spelling rules do not overtly mark word boundaries (as is the case in Japanese, Chinese or Thai); or because it corresponds to unsegmented phonetic transcripts. We will concentrate here on the

[28] And also in several other software packages, notably the parallel implementation of Giza in (Gao and Vogel, 2008).

[29] To give an idea of the computational cost, aligning the French/English Hansard corpus in both directions takes approximately a day. This corpus contains over a million sentence pairs extracted from the proceedings of the Canadian Parliament.

latter situation, which is closer to our scenario, but similar ideas could be used (and have been used) in the former (see eg. (Xu et al., 2008)).

The most obvious approach, that we first review, consists in using segmentation or, more generally, morphological analysis, as a preprocessing step before computing alignments. This strategy is mostly implemented with the Machine Translation task in mind. The second part of this section then focusses on models learning jointly morphology and alignments. If the influence of Machine Translation is still pervading, these models also enable for instance the extraction of a bilingual lexicon and the identification of morpheme boundaries in one of the languages in the bilingual pair. In other words, as segmentation is often used to alleviate the difficulty to align words, alignments can also be simultaneously used to guide segmentation.

4.1 Segment, then align

The most obvious way to reconcile the source and target side notions of words is to preprocess the morphologically rich language so as to decompose complex lexical forms into shorter segments, and/or to neutralize morphological variations that are not marked (and thus not necessary for the translation process) in the morphologically simpler one: forms that only differ in their case mark can, for instance, be collapsed into one non-marked version for the purpose of translating into English, where case is not marked.

This strategy has been successfully applied to many language pairs in the context of Machine Translation applications: (Nießen and Ney, 2001) is a first attempt to cluster morphological variants when translating from German into English; while (Koehn and Knight, 2003) and (Dyer, 2009) are early attempts at splitting German compounds (see also (Durgar El-Kahlout and Yvon, 2010; Fraser et al., 2012) for more recent studies of translation out of German). Fishel and Kirik (2010) also adopt this trend of research as they use unsupervised morphological analysis on an Estonian-English corpus so as to perform alignment on several variants of the lemmatized Estonian part of the corpus. Similar techniques have been proposed for other language pairs such as for instance Czech (Goldwater and McClosky, 2005), Arabic (Habash and Sadat, 2006), Spanish (de Gispert and Mariño, 2008), Finnish (Virpioja et al., 2007), Turkish (Oflazer and El-Kahlout, 2007) to name a few early studies. Note that the reverse approach (eg. splicing English words into complex forms) has also been attempted eg. in (Ueffing and Ney, 2003). In any case, the benefits (in terms of translation quality) of such preprocessing can be limited, except for the translation of out-of-vocabulary words. An important methodological issue here is to find formal criteria for choosing the best segmentation and clustering strategy: for lack of casting these choices into a well defined optimization problem, most current approaches resort to heuristic, trial-and-error procedures oriented by the final BLEU score.

Such preprocessing typically implies external resources and tools for morphological analysis in the source and/or the target language, but can also employ, in rarer cases, the unsupervised segmentation techniques introduced in Section 2: this is for instance the case of (Virpioja et al., 2007).[30] As noted by several authors, decompos-

[30] A similar approach is developed for the related task of bilingual lexicon acquisition by Besacier et al.

ing word forms into morphemes goes against the main intuition of phrase-based SMT (Koehn, 2010), which favors the translation of large units; and also reduces the effectiveness of language models, as it decreases the size of the context. To mitigate these potentially negative effects, several authors have proposed to simultaneously consider multiple decomposition schemes, which are then recombined using system combination techniques (eg. Minimum Bayes Risk decoding) as in (Dyer, 2007) (for German), (de Gispert et al., 2009) (for Finnish) and (Virpioja et al., 2010) for German and Czech. In this way, it is possible to get the benefits of using large units in translation, when they are found in the training data, while still being able to produce unseen forms through morphological decomposition.

Another pitfall of this strategy is its linguistic bluntness: posterior to splitting, morphs of the same words behave as if they were completely unrelated, and can align to arbitrarily remote units in the other language; it is likewise impossible to control the fertility of alignments at the level of words. The model of Eyigöz et al. (2013) is intended to address these issues and develops a hierarchical view of alignment, which reintroduces the distinction between words and morphs. Assuming the availability of a morphological decomposition in both source and target, their model extends IBM Model 1 (and HMM) so as to constraint morpheme alignments with word alignments (if a source morph aligns with a target morph, then the corresponding word forms must also align. From a technical viewpoint their starting point is a two-level IBM model 1, where word alignments decompose into morpheme alignments.

If these approaches provide practical means to improve alignments involving one (or two) Morphologically Rich Languages (MRLs) – even more so when the Morphologically Poor Language (MPL) is English – they remain unsatisfactory, notably due to the lack of modeling the dependency between units in the source and the target side: it seems however reasonable to assume that the optimal segmentation / normalization of a given source language must depend on the target language it must align to. Joint models of segmentation and alignment, introduced below (§ 4.2), are intended to address this problem.

4.2 Joint models

Just like unsupervised segmentation models, joint models have been simultaneously developed for two separate applications: one is the automatic construction of bilingual dictionaries from parallel texts, where one side is an unsegmented symbol stream, and the other corresponds to regular words; the second is more related to machine translation applications. We start with the former.

4.2.1 Extracting bilingual dictionaries

The simplest way to go is just to use IBM models and their standard implementation using a maximal decomposition of the source into symbols (using for instance letters, or characters for Chinese). Assuming that word boundaries are observed on the target side, this approach amounts to replacing the term $t(e|f)$ in the conventional models

(2006); Stüker et al. (2009) (see also section 4.2 below).

with terms such as $\prod_i t(f_i|f)$ or $\prod_i t(f_i|e_{i-1}, f)$: instead of having target words f generate source words, they generate strings of source characters. One way to look at this extension is to represent the model as a HMM with one state for each target word, where each state emits individual characters, and either loops or transitions to another word.

Stahlberg et al. (2012, 2014) also consider the problem of aligning a sequence of phones / letters on the source side with a sequence of words in the target side. Even though the goal is not so clearly stated (extracting a bilingual dictionary ?), alignment here is asymmetric from target to source: each symbol in the source aligns with exactly one target word. The authors develop a variant of IBM model 3 where the lexical generation phase (for each target position, draw a symbol given the corresponding target word) is decomposed as: (a) draw the number of source symbols conditioned on the target; and (b) iteratively and independently draw each symbol. Having a two-level generation process that explicitly identifies words is an improvement over the previous model.

4.2.2 Joint alignment and morphological segmentation

In this section, we briefly review similar attempts to develop joint models with machine translation applications in mind. Note that in a complete MT setting, the problem is slightly more complex that when just alignment is targeted, as it requires to also segment the source even if the target is unknown (in decoding). We start with asymmetric approaches, where the segmentation in words of one side of the bi-text is known and kept fixed; we then briefly review more ambitious attempts to learn the segmentation simultaneously on both sides.

Asymmetric approaches An early joint model is the proposal of Deng and Byrne (2008), which extends the HMM model of Vogel et al. (1996) by allowing a target state to generate multiple words on the source side. The authors develop an analogous to IBM Model 4, where the fertility of target words is explicitly controled, while preserving the desirable properties of the HMM model (efficient decoding algorithm, exact computation of posteriors, well-understood learning procedure, etc). In this approach, an alignment is composed of a set of random variables $\{(a, h, \phi)_k, k = 1 \dots K\}$ where a_k is the index of the target equivalent of the kth source phrase, h_k a binary indicator for null alignment (when $h_k = 0$, the target word is in fact empty), and ϕ_k the length of the k^{th} phrase. Given these, the alignment probability of the k^{th} phrase $P(v_k|a_k, h_k, \phi_k, \mathbf{e})$ is a mere product over the words $f_{1,k} ... f_{\phi_k,k}$ in v_k of the standard translation model parameters $t(f_{i,k}|e_{a_k})$. Assuming Markovian dependence assumptions between a_k as well as other simplifiying hypotheses regarding h_k (a fix insertion rate) and ϕ_k (only depends on e_{a_k}), this model can be trained efficiently. Another variant is also considered, where a bigram model is used for $P(v_k|a_k, h_k, \phi_k, \mathbf{e})$. The applications studied in this paper are phrase extraction and scoring on the one hand, automatic alignment of Chinese to English: note that joint segmentation and alignment of the target would also be readily possible.

Among the early approaches using bilingual information to segment text, Xu et al. (2008) present a Bayesian model able to learn a Chinese text segmentation suitable for

Machine Translation. It assumes that the corpus of parallel sentences (c_1^K, e_1^I), with c_1^K a sequence of Chinese characters and e_1^I a sequence of English words, is generated in parallel to a hidden sequence of Chinese words f_1^J and a hidden sequence of word alignments b_1^I. The joint probability of a sentence pair and its hidden variables then factorizes to:

$$P(c_1^K, e_1^I, f_1^J, b_1^I) = P(f_1^J)\delta(f_1^J, c_1^K)P(e_1^I, b_1^I|f_1^J).$$

$P(f_1^J)$ is specified by the monolingual unigram word model of (Goldwater et al., 2006a) (see Section 2.5.1) with a slightly modified spelling model incorporating a Poisson prior on word length, and the translation probability $P(e_1^I, b_1^I|f_1^J)$ is specified by the IBM model 1, modified so that a Dirichlet Process prior is placed on the distributions over English words depending on the Chinese word it is aligned to. The translation probability hence rewrites as:

$$P(e_1^I, b_1^I|f_1^J) = \frac{1}{(J+1)^I}\prod_i^I P(e_i|G_{f_{b_i}})$$

with $G_{f_{b_i}} \sim DP(\alpha, P_0(e))$ (see appendix A.1.3), and $P_0(e)$ the empirical distribution over English words in the data. The final model involves an equivalent factor for the other direction (English to Chinese), with a subsequent weighting of both components. Inference is performed using a Gibbs sampler considering only the alignment hypotheses that are close to the current alignment. Eventually, the Gibbs sampling procedure is combined iteratively with a realignment step using the `Giza++` toolkit (Och and Ney, 2003), improving performance.

The approach of Chung and Gildea (2009) is pretty similar to (Deng and Byrne, 2008): the (unsegmented) target is generated by a semi-Markov model akin to the IBM model 1, where each English word can generate any string of target characters (up to a maximum length). A prior on length is also necessary to favor small segments, and a variational Bayes approach similar to (Johnson, 2007) is adopted to mitigate the overfitting effect of EM. This model is contrasted with the simplest of all monolingual models (a one-state semi-HMM, see above in section 2.2).

Fishel (2009) considers an alignment between morphemes on the (highly inflectional) source side and words on the target side, with a notion of consistency (target morphemes cannot intersect and need to cover the whole source sentence). This notion of consistency is reminiscent, albeit more loosely specified, of the one encountered in (Eyigöz et al., 2013). The method of (Liang et al., 2006) serves as an inspiration for jointly maximizing the alignment probabilities in both direction although the consistency constraint requires the implementation of a rather ad-hoc adaptation of the forward-backward algorithm. The theoretical implications of that procedure seem in fact rather hard to evaluate. Results also prove to be below a baseline (an out-of-the-box system without any segmentation) when this method is evaluated as a preprocessing segmentation step for a translation system.

The work of Naradowsky and Toutanova (2011) presents another, somewhat more ambitious, extension of conventional word alignment models. The starting point is again the HMM model of Vogel et al. (1996), which is improved in many different ways:

1. the source is a sequence of morphemes; source states are pairs $y = [a, t]$ where a is the index of the source morpheme (in $[0 : J]$) and t is a restriction on the emitted target morpheme (a prefix, a suffix, or a stem). The authors propose to use a log-linear parameterization à la Berg-Kirkpatrick et al. (2010) which enables them to include rich-features (here: dependency and POS information) on top of the conventional distortion-based model;
2. source states emit target morphemes; the corresponding distribution includes a first order dependency over past morphemes, and an additional conditioning over word boundary indicators. Estimation is based on smoothed frequency ratio;
3. word boundaries in the target also need to be accounted for, even if they are observed: this means that HMM states truly emit (a) a word boundary Bernoulli variable (with Markovian dependencies) and (b) a target morpheme;

Denoting $\mathbf{ta}$ the generalized alignment vector, $\boldsymbol{\mu}$ the target morpheme sequence, and $\mathbf{b}$ the word boundaries variables ($b_i = 1$ at word endings), the overal model defines $P(\boldsymbol{\mu}, \mathbf{b}, \mathbf{ta}|\mathbf{f})$ as a product of three terms: the first accounts for the translation model and is essentially a product over morphemes of terms $P(\mu_i|ta_i, \mathbf{f})$; the second is a usual distortion model $P(ta_i|ta_{i-1}, \mathbf{f})$; and the third one is needed to model word endings $P(b_i|\mathbf{f}, b_{i-1})$.

Training is performed using EM: the E step computes the posteriors, and the M step computes the expectation of counts (for the emission models) then reestimates the transition kernel using LBFGS. Note that the whole model looks somewhat overparameterized: a "prefix" state should not emit a word boundary; and should not be followed by a "suffix" state.

The work of Nguyen et al. (2010) can be viewed as a bilingual version of (Mochihashi et al., 2009) presented in Section 2.5.1: in this version, a pair of unsegmented (source) and segmented (target) sentences is generated in two steps: first an unsupervised segmentation (using the Nested model of (Mochihashi et al., 2009)), then an alignment (conditional to the source segmentation), where the alignment is made of unaligned source words, unaligned target words, and one-to-one word pairs. Inference is performed via sampling segmentation points (taking the alignment into account): note that to speed-up learning, likely segmentation points in the source are pre-computed with a rule-based system. This means that this approach essentially learns to join in an optimal fashion (wrt alignment) the pre-segmented morphemes in the source. This model is used in MT, where the optimal source segmentation is computed based on the unsupervised segmentation component only.

Symmetric approaches We now turn to approaches aimed at simultaneously segmenting the source and the target side. An early line of work following this path can be identified in attempts to directly extract bilingual phrases for phrase-based SMT, instead of relying on the usual heuristic pipeline (Koehn, 2010).

A first model is introduced in (Marcu and Wong, 2002), where a pair $(\mathbf{f}, \mathbf{e})$ of sentences is generated in two steps: (a) generate K hidden concepts, each generating in turn a phrase-pair (f, e), and (b) reorder the phrases on each side to recover $(\mathbf{f}, \mathbf{e})$. Note that this, as well as many other approaches along these lines, prevents to extract discontinuous phrases. Two models of increasing complexity are considered, depending on how they model the reordering component; in both cases estimation is computationally challenging and requires both to heavily filter the repertoire of possible phrase pairs, and to develop approximate estimation techniques. A conditional version (eg. asymmetric) of this model, akin to an IBM Model 3 version for phrases, is in (DeNero et al., 2006): learning its parameter with EM is however untractable, due to the need to sum over all segmentations and alignments, and is approximated here using constraints derived from word alignments. As noted by DeNero et al. (2008), these approaches are plagued by degenerate solutions, corresponding to a clear tendency to under-segment the corpus. These authors propose to introduce priors in order to constraint the model in principled ways, and develop an algorithm based on Gibbs sampling to compute count expectations over the posterior distribution of latent segmentation and alignment variables, while taking such prior information into account: sampling considers various simple operators to move from one assignment of these variables into another. Upon convergence, the expectations can be plugged into the M-Step of the EM algorithm to readily derive phrase translation probabilities.

Note that similar techniques lie at the core of the work reported in (Snyder and Barzilay, 2008a,b), which however consider the segmentation of words in character substrings, rather than of sentences into phrases. In this model, parallel phrases are obtained by sampling from two monolingual distribution of morphemes, plus one bilingual distribution over abstract morphemes. This model is based on the Dirichlet process, and can also integrate prior information regarding abstract morphemes (for instance using string similarity when the two languages are orthographically or phonetically related).

Almost simultaneously, a series of papers (Zhang et al., 2003; Zhang and Vogel, 2005a; Vogel, 2005; Zhang and Vogel, 2005b) develops an alternative approach to phrase alignments, which uses phrase-to-phrase association scores, such as the Pointwise Mutual Information or aggregates derived from IBM Model 1 scores, instead of a sound probabilistic model. With the exception of (Zhang et al., 2003), where a phrasal alignment is actually built, these scores are mostly used to perform phrase extraction on the fly for Machine Translation. These attempts have been continued in (Xu et al., 2006), and more recently in (Lardilleux et al., 2012; Gong et al., 2013). Note that they start with association scores attached to minimal units, a requirement that is difficult to meet with unsegmented character (or phonemic) strings.

Phrase-to-phrase alignments are also studied in Neubig et al. (2011), where the authors use Bayesian non-parametric techniques on top of ITG alignments (Wu, 1997) to extract many-to-many phrasal alignments *with varying levels of granularity*, thereby fixing a well-known issue with phrasal-alignment models, which typically lack the ability that heuristic approaches have to extract small units embedded within larger units. This work continues in Neubig et al. (2012), where the focus shifts from the alignment of supra-word units to infra-word units: the resulting alignment of variable length character strings is then used to train a character-based translation model, thereby mitigating the data sparsity issues faced with systems operating at the level of words.

4.3 Summary

In this last section, we have reviewed various attempts to integrate unsupervised morphology learning techniques with unsupervised alignment techniques initially developed for Machine Translation systems. This trend of research has become increasingly active in the past years, moving from strategies using segmentation as a preprocessing step before alignments are performed, to models aiming at learning jointly relevant segmentation and alignment. Adams et al. (2015) report performance improvements for the latter approach on a bilingual lexicon induction task, with the additional benefit of achieving high precision even on a very small corpus, which is of particular interest in the BULB project's context. More work could be done in order to, for example, take tones into account or use target words to control the length of source units.

5 General conclusion

In this document, we reviewed the literature studying, on one hand, the unsupervised learning of morphology in a monolingual context, and on the other hand, the induction of cross-lingual alignments. We then focussed on more recent efforts to combine these techniques in a joint learning framework, allowing in particular to cope with small quantities of data, as we are dealing with this particular constraint in the BULB project.

Many questions still need to be addressed. Implicit choices are made through the way data are specified and represented. Most of the work we studied consider character strings as their inputs. Taking for example tones into account, prosodic markers, or even a partial bilingual dictionary, would require different kinds of input data, and the development of models able to take advantage of this additional information.

A second observation is that a large part of the work considered here aims at injecting some sort of prior knowledge about the desired form of the linguistic units we want to extract and relate to others. Most EM training schemes deployed in this literature tend to otherwise produce degenerated and trivial (over-segmented or conversely unsegmented) solutions. Hypotheses made to control this phenomenon are likely to impact largely the nature of the units that we isolate and identify. The BULB project aims at supporting an effort to document endangered languages and we should consequently question as systematically as possible the linguistic validity of those hypotheses and the results they produce. This is rarely done ; the question of the validity of the units considered is often subjected to a particular task's evaluation (eg. Machine Translation or segmentation into morphemes). The Adaptor Grammar framework (Johnson et al., 2007a), which enables the specification of high level linguistic hypotheses which can then be confronted with the data, is a rare occurrence in the body of literature we reviewed.

More generally, a careful inventory of priors derived from the linguistic knowledge at our disposal should be undertaken. This is especially true regarding cross-lingual priors we can postulate about French on one hand, and Basaa, Myene and Embosi on the other end. There is indeed a paradox in the literature we reviewed, since a certain fantasy of generality underlying most unsupervised learning approaches seems to often conflict with the feeling that the methods developed are in fact quite dependent to the language, or the language pair, studied in those works.

A last large group of questions concerns the statistical methods used during inference. While MCMC sampling methods seem to avoid certain problems encountered with the EM algorithm, they also tend to be heavy computationally and sometimes likely to converge very slowly. Ad-hoc, for example "blocked", sampling methods must then be developed to tackle particular problems. Another important aspect of the task we are facing lies in the noisy nature of the input we will inherit from the phonemicization of the unwritten language. Processing a phoneme lattice instead of a phonemic transcription, following the work of Neubig et al. (2010), seems an extremely relevant strategy in our context.

The field of research to which we tried to provide an exhaustive review in this document is extremely active and prone to constant evolution. As with many Natural Language Processing challenges, models and algorithms can become quite involved despite the rather shallowness of their linguistic substance. This is certainly an intimidating, yet exciting, aspect of the task at hand.

A Bayesian non-parametric

A.1 Basics

A.1.1 Chinese Restaurant Process

The simplest model is the so-called *Chinese Restaurant Process* (CRP), which generates partitions of integers. The analogy goes as follows: each customer (represented as an integer) from $\{1 \dots T\}$ sequentially enters a restaurant with infinite number of tables, each table accommodating an infinite number of customers. When customer t enters, an arrangement $\boldsymbol{z}^{-t}$ of the previous customers is observed, with m non-empty tables each accommodating $n_k(\boldsymbol{z}^{-t})$ customers. The customer either seats at a non-empty table with probability $P(z_t = k|\boldsymbol{z}^{-t})$ or chooses a new one with probability $P(z_t = m+1|\boldsymbol{z}^{-t})$. These terms are defined as follows:

$$P(z_t = k|\boldsymbol{z}^{-t}) = \begin{cases} \frac{n_k(\boldsymbol{z}^{-t})}{t-1+\alpha} & \text{if } 1 \leq k \leq m \\ \frac{\alpha}{t-1+\alpha} & \text{if } k = m+1 \end{cases}$$

with $\alpha \geq 0$ a parameter of the process called the concentration [31] parameter. Larger values for this parameter result in a tendency towards the opening of more new tables, hence a more uniform distribution of customers across the tables. The probability of a given sequence of table assignments $\boldsymbol{z}$ for n customers is given by:

$$P(\boldsymbol{z}) = \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \cdot \alpha^K \cdot \prod_{k=1}^{K} (n_k - 1)!$$

with K the total number of tables in the arrangement $\boldsymbol{z}$, and n_k the number of customers at table k. Γ is the Gamma function defined by $\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du$ for $x > 0$.

A.1.2 Pitman-Yor process

The Pitman-Yor process is a generalization of the Chinese Restaurant Process, in which the conditional probability for the kth customer is defined by:

$$P(z_t = k|\boldsymbol{z}^{-t}) = \begin{cases} \frac{n_k(\boldsymbol{z}^{-t})-\beta}{t-1+\alpha} & \text{if } 1 \leq k \leq m \\ \frac{m\beta+\alpha}{t-1+\alpha} & \text{if } k = m+1 \end{cases}$$

with $\alpha \geq 0$ and $0 \leq \beta < 1$ two parameters of the model, and m the number of tables already occupied when the kth customer enters the restaurant. The first parameter is the equivalent to the concentration parameter in the CRP process while β allows to have more control over the shape of the tail of the distribution.

[31] Maybe the term *dispersion*, sometimes used in place of *concentration*, would be more appropriate since higher values of this parameter lead to a higher dispersion of the customers across the tables.

A.1.3 Dirichlet Process

A Dirichlet Process $\mathrm{DP}(\alpha, G_0)$, with a concentration parameter α and a base distribution G_0, is a stochastic process whose sample path is a probability distribution over a measurable set S. For any partition $B_1, \ldots, B_n$ of S, if $X \sim \mathrm{DP}(\alpha, G_0)$ then

$$(X(B_1), \ldots, X(B_n)) \sim \mathrm{Dir}(\alpha G_0(B_1), \ldots, \alpha G_0(B_n))$$

where $\mathrm{Dir}(\cdot)$ is the Dirichlet distribution.

A "two-stage" CRP model, in which customers are seated according to a certain concentration parameter α, and each new opened table is labelled with a draw from a distribution G_0, is equivalent to a Dirichlet process with concentration parameter α and base distribution G_0.

A.2 Sampling

Bayesian approaches are interested in taking into account the full posterior distributions for the parameters of the model instead of a point-wise estimate (like MAP or ML estimates) usually obtained via an EM procedure. Since the posterior distribution is usually impossible to express analytically, sampling algorithms known as Markov Chain Monte Carlo (MCMC) methods are used.

A.2.1 MCMC methods

Following (Goldwater, 2006), we give the main ideas these methods are built upon.

— A markov chain with random states $Y^1 \ldots Y^T$ and transition matrix $\mathbf{P}$ is built with proper conditions ensuring its convergence to a unique stationary distribution π over states satisfying $\pi\mathbf{P} = \pi$.

— states of the Markov chain correspond to an assignment of values to the parameters we want to sample from. The state space of the Markov chain corresponds to the hypothesis space of the model.

— Proper construction of $\mathbf{P}$ guarantees π will be the distribution we are interested to infer. After convergence, each Y^t will be a sample of the distribution of interest. The conditions on the Markov chain are the following:

  — the chain needs to be irreducible (existence of a finite path with non-zero probability between all pairs of states),

  — the chain also needs to be aperiodic (together with the preceding condition, this defines an "ergodic" chain),

  — $\mathbf{P}$ needs to satisfy $\pi\mathbf{P} = \pi$ when π is the distribution we want to sample from. This is the "general balance" condition.

A particular algorithm build on these principles is Gibbs sampling. If we decompose each state variable Y^t into its K components $Y_1^t \ldots Y_K^t$ corresponding to different variables in the model, each iteration of this sampler corresponds to K steps. In each of these steps, the kth component Y_k^t is sampled from its conditional distribution given the current values of all the other components.

To guarantee ergodicity, we need to avoid cases where the conditional probabilities take null values – for example when a variable is tied to another variable in a manner that changing only the first one produces a state with zero probability. To address this problem, it is possible to "block" a Gibbs sampler, sampling a block of variables at once instead of separately, which can also improve convergence speed. The effort to accelerate the convergence also leads sometimes to the use of a technique known as *simulated annealing*.[32]

A.2.2 Exchangeability

A sequence of random variables $Z_1, \ldots, Z_n$ is exchangeable if the joint probability of the sequence is not changed by a permutation of the indices of the sequence, in other words if, for any permutation σ,

$$P(Z_1, Z_2, \ldots, Z_n) = P(Z_{\sigma(1)}, Z_{\sigma(2)} \ldots, Z_{\sigma(n)}).$$

Note that exchangeability is related, but distinct, to the concept of a series of independant and identically distributed (i.i.d.) random variables.[33]

A sequence of variables distributed according to a distribution itself drawn from a Dirichlet Process has this property of exchangeability, which is crucial to perform inference using Gibbs sampling efficiently: any assignment of a component can be made under the assumption that this component is the last one in the sequence. This way, one can avoid recomputing counts for the part of the sequence occurring after the currently assigned component.

[32] At the beginning of the sampling, low-probability choices are encouraged so as to allow the sampler to explore more rapidly a larger area of the search space.

[33] De Finetti's theorem states that exchangeable observations are conditionally independent given some latent variable.

References

Adams, O., Neubig, G., Cohn, T., and Bird, S. (2015). Inducing Bilingual Lexicons from Small Quantities of Sentence-Aligned Phonemic Transcriptions. In *12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam.

Aronoff, M. (1976). *Word Formation in Generative Grammar*. Linguistic inquiry monographs. MIT Press, Cambridge, MA.

Ayan, N. F. and Dorr, B. J. (2006). A maximum entropy approach to combining word alignments. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 96–103, New York City, USA. Association for Computational Linguistics.

Berg-Kirkpatrick, T., Bouchard-Côté, A., DeNero, J., and Klein, D. (2010). Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California. Association for Computational Linguistics.

Besacier, L., Zhou, B., and Gao, Y. (2006). Towards speech translation of non written languages. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 222–225. IEEE.

Bimbot, F., Deligne, S., and Yvon, F. (1995). Unsupervised decomposition of phoneme strings into variable-length sequences, by multigrams. In *International Conference of PHonetic Sciences (ICPHS)*, Stockholm, Sweden.

Bird, S. (2011). Bootstrapping the language archive: New prospects for natural language processing in preserving linguistic heritage. *Linguistic Issues in Language Technology*, 6:1–16.

Bird, S. and Chiang, D. (2012). Machine translation for language preservation. In *Proceedings of COLING 2012: Posters*, pages 125–134, Mumbai, India. The COLING 2012 Organizing Committee.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2002). Latent Dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 601–608.

Blevins, J. P. (2006). Word-based morphology. *Journal of Linguistics*, 42(03):531–573.

Blitzer, J., Globerson, A., and Pereira, F. (2005). Distributed latent variable models of lexical co-occurrences. In *In the Proceedings of the International Workshop on Artificial Intelligence and Statistics*, AISTATS, page 10.

Blunsom, P. and Cohn, T. (2006). Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 65–72, Sydney, Australia.

Börschinger, B. and Johnson, M. (2011). A particle filter algorithm for bayesian word segmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia.

Börschinger, B. and Johnson, M. (2012). Using rejuvenation to improve particle filtering for bayesian word segmentation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–89, Jeju Island, Korea. Association for Computational Linguistics.

Botha, J. A. and Blunsom, P. (2013). Adaptor grammars for learning non-concatenative morphology. In *EMNLP*, pages 345–356.

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867.

Brent, M. R. and Cartwright, T. A. (1996). Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125.

Brown, P. F., Cocke, J., Pietra, S. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chan, E. (2006). Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology at HLT-NAACL 2006*, pages 69–78, New York City, USA. Association for Computational Linguistics.

Chung, T. and Gildea, D. (2009). Unsupervised tokenization for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 718–726, Singapore. Association for Computational Linguistics.

Cohen, S. B., Blei, D. M., and Smith, N. A. (2010). Variational inference for adaptor grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 564–572. Association for Computational Linguistics.

Cotterell, R., Peng, N., and Eisner, J. (2015). Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3:433–447.

Creutz, M. (2003). Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 280–287, Sapporo, Japan. Association for Computational Linguistics.

Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Creutz, M. and Lagus, K. (2004). Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pages 43–51, Barcelona, Spain. Association for Computational Linguistics.

Creutz, M. and Lagus, K. (2005). Inducing the morphological lexicon of a natural language from unannotated text. In *In Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05*.

Creutz, M. and Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34.

de Gispert, A. and Mariño, J. B. (2008). On the impact of morphology in english to spanish statistical mt. *Speech Commununication*, 50(11-12):1034–1046.

de Gispert, A., Virpioja, S., Kurimo, M., and Byrne, W. (2009). Minimum bayes risk combination of translation hypotheses from alternative morphological decompositions. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 73–76, Boulder, Colorado.

de Marcken, C. (1996a). Linguistic structure as composition and perturbation. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 335–341. Association for Computational Linguistics.

de Marcken, C. (1996b). *Unsupervised Language Acquisition*. PhD thesis, Department of Computer Science, MIT.

De Pauw, G. and Wagacha, P. W. (2007). Bootstrapping morphological analysis of Gĩkũyũ using unsupervised maximum entropy learning. In Van hamme, H. and van Son, R., editors, *Proceedings of the Eighth Annual Conference of the International Speech Communication Association*, Antwerp, Belgium.

Déjean, H. (1998). Morphemes as necessary concept for structures discovery from untagged corpora. In *Proceedings of the Workshop on Paradigms and Grounding in Natural Language Learning*, pages 295–299, Adelaide, Australia.

Deligne, S. and Bimbot, F. (1995). Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 169–172. IEEE.

Deligne, S. and Bimbot, F. (1997). Inference of variable-length linguistic and acoustic units by multigrams. *Speech Communication*, 23(3):223–241.

Deligne, S., Yvon, F., and Bimbot, F. (1995). Variable-length sequence matching for phonetic transcription using joint multigrams. In *Fourth European Conference on Speech Communication and Technology*.

Deligne, S., Yvon, F., and Bimbot, F. (2001). Selection of multiphone synthesis units and grapheme-to-phoneme transcription using variable-length modeling of strings. In *Data-Driven Techniques in Speech Synthesis*, pages 125–147. Springer.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

DeNero, J., Bouchard-Côté, A., and Klein, D. (2008). Sampling alignment structure under a bayesian translation model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 314–323.

DeNero, J., Gillick, D., Zhang, J., and Klein, D. (2006). Why generative phrase models underperform surface heuristics. In *Proceedings of the ACL workshop on Statistical Machine Translation*, pages 31–38, New York City, NY.

DeNero, J. and Macherey, K. (2011). Model-based aligner combination using dual decomposition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 420–429, Portland, Oregon, USA. Association for Computational Linguistics.

Deng, Y. and Byrne, W. J. (2008). HMM word and phrase alignment for statistical machine translation. *IEEE Transactions on Audio, Speech & Language Processing*, 16(3):494–507.

Dreyer, M. and Eisner, J. (2009). Graphical models over multiple strings. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 101–110, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dreyer, M. and Eisner, J. (2011). Discovering morphological paradigms from plain text using a dirichlet process mixture model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 616–627, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dreyer, M., Smith, J. R., and Eisner, J. (2008). Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1080–1089, Stroudsburg, PA, USA. Association for Computational Linguistics.

Durgar El-Kahlout, I. and Yvon, F. (2010). The pay-offs of preprocessing for German-English Statistical Machine Translation. In Federico, M., Lane, I., Paul, M., and Yvon, F., editors, *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 251–258.

Dyer, C. (2009). Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 406–414, Boulder, Colorado. Association for Computational Linguistics.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Dyer, C. J. (2007). The "noisier channel": Translation from morphologically complex languages. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 207–211, Prague, Czech Republic. Association for Computational Linguistics.

Eifring, H. and Theil, R. (2004). Linguistics for students of Asian and African languages. *Institutt for østeuropeiske og orientalske studier*.

Eyigöz, E., Gildea, D., and Oflazer, K. (2013). Simultaneous word-morpheme alignment for statistical machine translation. In *HLT-NAACL*, pages 32–40.

Fishel, M. (2009). Deeper than words: Morph-based alignment for statistical machine translation. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics PacLing*, PACLING'09.

Fishel, M. and Kirik, H. (2010). Linguistically motivated unsupervised segmentation for machine translation. In *Proceedings of the Language Ressources and Evaluation Conference*, La Valette, Malta.

Fraser, A. and Marcu, D. (2007). Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.

Fraser, A., Weller, M., Cahill, A., and Cap, F. (2012). Modeling inflection and word-formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 664–674, Avignon, France. Association for Computational Linguistics.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2006a). Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney, Australia. Association for Computational Linguistics.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2006b). Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, pages 459–466, Cambridge, MA. MIT Press.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2009). A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Goldwater, S. and McClosky, D. (2005). Improving statistical mt through morphological analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 676–683, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Goldwater, S. J. (2006). *Nonparametric Bayesian models of lexical acquisition*. PhD thesis, Citeseer.

Gong, L., Max, A., and Yvon, F. (2013). Improving bilingual sub-sentential alignment by sampling-based transpotting. In *Proceedings of the International Workshop on Spoken Language Translation*, page 8 pages, Heidelberg, Germany.

Graça, J., Ganchev, K., and Taskar, B. (2010). Learning tractable word alignment models with complex constraints. *Computational Linguistics*, 36:481–504.

Grönroos, S.-A., Virpioja, S., Smit, P., and Kurimo, M. (2014). Morfessor flatcat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Habash, N. and Sadat, F. (2006). Arabic preprocessing schemes for statistical machine translation. In *NAACL '06: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pages 49–52, Morristown, NJ, USA. Association for Computational Linguistics.

Hammarström, H. and Borin, L. (2011). Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Harris, Z. S. (1955). From Phoneme to Morpheme. *Language*, 31(2):pp. 190–222.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1):177–196.

Hofmann, T. and Puzicha, J. (1998). Statistical models for co-occurrence data. Technical report, Massachussets Institute of Technology, Artificial Intelligence Laboratory.

Hu, Y., Matveeva, I., Goldsmith, J., and Sprague, C. (2005). Refining the SED heuristic for morpheme discovery: Another look at Swahili. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 28–35, Ann Arbor, Michigan. Association for Computational Linguistics.

Johnson, M. (2007). Why doesn't em find good hmm pos-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic. Association for Computational Linguistics.

Johnson, M. (2008a). Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27, Columbus, Ohio. Association for Computational Linguistics.

Johnson, M. (2008b). Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure. In *Proceedings of ACL-08: HLT*, pages 398–406, Columbus, Ohio. Association for Computational Linguistics.

Johnson, M., Christophe, A., Dupoux, E., and Demuth, K. (2014). Modelling function words improves unsupervised word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 282–292, Baltimore, Maryland. Association for Computational Linguistics.

Johnson, M. and Goldwater, S. (2009). Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325, Boulder, Colorado. Association for Computational Linguistics.

Johnson, M., Griffiths, T. L., and Goldwater, S. (2007a). Adaptor grammars: a framework for specifying compositional nonparametric bayesian models. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 641–648, Cambridge, MA. MIT Press.

Johnson, M., Griffiths, T. L., and Goldwater, S. (2007b). Bayesian inference for PCFGs via markov chain monte carlo. In *HLT-NAACL*, pages 139–146.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P. and Knight, K. (2003). Empirical methods for compound splitting. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 187–193, Morristown, NJ, USA. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistic*, pages 127–133, Edmondton, Canada.

Kohonen, O., Virpioja, S., and Lagus, K. (2010). Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.

Lardilleux, A., Yvon, F., and Lepage, Y. (2012). Hierarchical sub-sentential alignment with Anymalign. pages 279–286, Trento, Italy.

Liang, P., Taskar, B., and Klein, D. (2006). Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA. Association for Computational Linguistics.

Marcu, D. and Wong, D. (2002). A phrase-based,joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139.

Mochihashi, D., Yamada, T., and Ueda, N. (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-Volume 1*, pages 100–108. Association for Computational Linguistics.

Moore, R. C. (2005). A discriminative framework for bilingual word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Vancouver, British Columbia, Canada.

Naradowsky, J. and Toutanova, K. (2011). Unsupervised bilingual morpheme segmentation and alignment with context-rich hidden semi-markov models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 895–904, Portland, Oregon, USA. Association for Computational Linguistics.

Neubig, G., Mimura, M., Mori, S., and Kawahara, T. (2010). Learning a language model from continuous speech. In *INTERSPEECH*, pages 1053–1056. Citeseer.

Neubig, G., Watanabe, T., Mori, S., and Kawahara, T. (2012). Machine translation without words through substring alignment. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 165–174, Jeju Island, Korea. Association for Computational Linguistics.

Neubig, G., Watanabe, T., Sumita, E., Mori, S., and Kawahara, T. (2011). An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 632–641.

Nguyen, T., Vogel, S., and Smith, N. A. (2010). Nonparametric word segmentation for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 815–823, Stroudsburg, PA, USA. Association for Computational Linguistics.

Niehues, J. and Vogel, S. (2008). Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25, Columbus, Ohio.

Nießen, S. and Ney, H. (2001). Toward hierarchical models for statistical machine translation of inflected languages. In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*, pages 47–51, Toulouse, France.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

O'Donnell, T. J., Tenenbaum, J. B., and Goodman, N. D. (2009). Fragment grammars: Exploring computation and reuse in language. Technical report, Massachusetts Institute of Technology.

Oflazer, K. and El-Kahlout, I. D. (2007). Exploring different representational units in english-to-turkish statistical machine translation. In *StatMT '07: Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

Pearl, L., Goldwater, S., and Steyvers, M. (2010). Online learning mechanisms for bayesian models of word segmentation. *Research on Language and Computation*, 8(2-3):107–132.

Peng, N., Cotterell, R., and Eisner, J. (2015). Dual decomposition inference for graphical models over strings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 917–927, Lisbon, Portugal. Association for Computational Linguistics.

Rissanen, J. (1989). *Stochastic Complexity in Statistic Inquiry*. Series in Computer Science - Vol 15. World Scientific Publishing.

Simion, A., Collins, M., and Stein, C. (2015). On a strictly convex ibm model 1. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 221–226, Lisbon, Portugal. Association for Computational Linguistics.

Snyder, B. and Barzilay, R. (2008a). Cross-lingual propagation for morphological analysis. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 848–854. AAAI Press.

Snyder, B. and Barzilay, R. (2008b). Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio.

Stahlberg, F., Schlippe, T., Vogel, S., and Schultz, T. (2012). Word segmentation through cross-lingual word-to-phoneme alignment. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 85–90. IEEE.

Stahlberg, F., Schlippe, T., Vogel, S., and Schultz, T. (2014). Word segmentation and pronunciation extraction from phoneme sequences through cross-lingual word-to-phoneme alignment. *Computer Speech & Language*, pages –.

Stüker, S., Besacier, L., and Waibel, A. (2009). Human Translations Guided Language Discovery for ASR Systems. In *10th International Conference on Speech Science and Speech Technology (InterSpeech 2009)*, pages 1–4, Brighton (UK). Eurasip.

Synnaeve, G., Dautriche, I., Börschinger, B., Johnson, M., and Dupoux, E. (2014). Unsupervised word segmentation in context. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2326–2334, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Taskar, B., Lacoste-Julien, S., and Klein, D. (2005). A discriminative matching approach to word alignment. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80, Morristown, NJ, USA. Association for Computational Linguistics.

Teh, Y. W. (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.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Tiedemann, J. (2011). *Bitext Alignment*. Number 14 in Synthesis Lectures on Human Language Technologies, Graeme Hirst (ed). Morgan & Claypool Publishers.

Tomeh, N., Allauzen, A., and Yvon, F. (2014). Maximum-entropy word alignment and posterior-based phrase extraction for machine translation. *Machine Translation*, 28(1):19–56.

Toutanova, K. and Galley, M. (2011). Why initialization matters for ibm model 1: Multiple optima and non-strict convexity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 461–466, Portland, Oregon, USA. Association for Computational Linguistics.

Udupa, R. and Maji, H. K. (2006). Computational complexity of statistical machine translation. In *Proceedings of the Meeting of the European Chapter of the Association for Computational Linguistics*, pages 25–32, Trento, Italy.

Ueffing, N. and Ney, H. (2003). Using pos information for statistical machine translation into morphologically rich languages. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.

Virpioja, S., Väyrynen, J., Mansikkaniemi, A., and Kurimo, M. (2010). Applying morphological decompositions to statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 195–200, Uppsala, Sweden. Association for Computational Linguistics.

Virpioja, S., Väyrynen, J. J., Creutz, M., and Sadeniemi, M. (2007). Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *Proceedings of the Machine Translation Summit XI*, pages 491–498, Copenhagen, Denmark Copenhagen, Denmark.

Vogel, S. (2005). PESA: Phrase pair extraction as sentence splitting. In *Proceedings of the tenth Machine Translation Summit*, Phuket, Thailand.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics*, pages 836–841, Morristown, NJ, USA.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.

Wu, D. (2012). *Foundations of Text Alignment: Statistical Machine Translation Models from Bitexts to Bigrammars*. Theory and Decision Library C. Springer.

Xu, J., Gao, J., Toutanova, K., and Ney, H. (2008). Bayesian semi-supervised Chinese word segmentation for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1017–1024, Manchester, UK. Coling 2008 Organizing Committee.

Xu, J., Zens, R., and Ney, H. (2006). Partitioning parallel documents using binary segmentation. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '06, pages 78–85. Association for Computational Linguistics.

Zhai, K., Boyd-Graber, J., and Cohen, S. B. (2014). Online adaptor grammars with hybrid inference. *Transactions of the Association for Computational Linguistics*, 2:465–476.

Zhang, Y. and Vogel, S. (2005a). Competitive grouping in integrated phrase segmentation and alignment model. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, ParaText '05, pages 159–162, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhang, Y. and Vogel, S. (2005b). An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *Proceedings of the meeting of the European Association for Machine Translation (EAMT)*, pages 294–301.

Zhang, Y., Vogel, S., and Waibel, A. (2003). Integrated phrase segmentation and alignment algorithm for statistical machine translation. In *Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on*, pages 567–573. IEEE.