Investigating Connectivity and Consistency Criteria for Phrase Pair Extraction in Statistical Machine Translation

Spyros Martzoukos, Christophe Costa Florêncio and Christof Monz
Intelligent Systems Lab Amsterdam, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
{S.Martzoukos, C.Monz}@uva.nl, chriscostafl@gmail.com

Abstract

The consistency method has been established as the standard strategy for extracting high quality translation rules in statistical machine translation (SMT). However, no attention has been drawn to why this method is successful, other than empirical evidence. Using concepts from graph theory, we identify the relation between consistency and components of graphs that represent word-aligned sentence pairs. It can be shown that phrase pairs of interest to SMT form a sigma-algebra generated by components of such graphs. This construction is generalized by allowing segmented sentence pairs, which in turn gives rise to a phrase-based generative model. A by-product of this model is a derivation of probability mass functions for random partitions. These are realized as cases of constrained, biased sampling without replacement and we provide an exact formula for the probability of a segmentation of a sentence.

1 Introduction

A parallel corpus, i.e., a collection of sentences in a source and a target language, which are translations of each other, is a core ingredient of every SMT system. It serves the purpose of training data, i.e., data from which translation rules are extracted. In its most basic form, SMT does not require the parallel corpus to be annotated with linguistic information, and human supervision is thus restricted to the construction of the parallel corpus.

The extraction of translation rules is done by appropriately collecting statistics from the training data. The pioneering work of (Brown et al., 1993) identified the minimum assumptions that should be made in order to extract translation rules and developed the relevant models that made such extractions possible.

These models, known as IBM models, are based on standard machine learning techniques. Their output is a matrix of word alignments for each sentence pair in the training data. These word alignments provide the input for later approaches that construct phrase-level translation rules which may (Wu, 1997; Yamada and Knight, 2001) or may not (Och et al., 1999; Marcu and Wong, 2002) rely on linguistic information.

The method developed in (Och et al., 1999), known as the *consistency* method, is a simple yet effective method that has become the standard way of extracting (source, target)-pairs of phrases as translation rules. The development of consistency has been done entirely on empirical evidence and it has thus been termed a heuristic.

In this work we show that the method of (Och et al., 1999) actually encodes a particular type of structural information induced by the word alignment matrices. Moreover, we show that the way in which statistics are extracted from the associated phrase pairs is insufficient to describe the underlying structure.

Based on these findings we suggest a phrase-level model in the spirit of the IBM models. A key aspect of the model is that it identifies the most likely partitions, rather than alignment maps, associated with appropriately chosen segments of the training data. For that reason, we provide a *general* construction of probability mass functions for partitions and, in particular, an exact formula for the probability of a segmentation of a sentence.

Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13), pages 93–101,
Sofia, Bulgaria, August 9, 2013.

2 Definition of Consistency

In this section we provide the definition of consistency, which was introduced in (Och et al., 1999), refined in (Koehn et al., 2003), and we follow (Koehn, 2009) in our description. We start with some preliminary definitions.

Let $S = s_1...s_{|S|}$ be a source sentence, i.e., a string that consists of consecutive source words; each word s_i is drawn from a source language vocabulary and i indicates the position of the word in S. The operation of string *extraction* from the words of S is defined as the construction of the string $s = s_{i_1}...s_{i_n}$ from the words of S, with $1 \leq i_1 < ... < i_n \leq |S|$. If $i_1, ..., i_n$ are consecutive, which implies that s is a substring of S, then s is called a source phrase and we write $s \subseteq S$. As a shorthand we also write $s_{i_1}^{i_n}$ for the phrase $s_{i_1}...s_{i_n}$. Similar definitions apply to the target side and we denote by T, t_j and t a target sentence, word and phrase respectively.

Let $(S = s_1 s_2 ... s_{|S|},\ T = t_1 t_2 ... t_{|T|})$ be a sentence pair and let A denote the $|S| \times |T|$ matrix that encodes the existence/absence of word alignments in (S, T) as

$$A(i,j) = \begin{cases} 1, & \text{if } s_i \text{ and } t_j \text{ are aligned} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

for all $i = 1, ..., |S|$ and $j = 1, ..., |T|$. Unaligned words are allowed. A pair of strings $(s = s_{i_1}...s_{i_{|s|}},\ t = t_{j_1}...t_{j_{|t|}})$ that is extracted from (S, T) is termed *consistent* with A, if the following conditions are satisfied:

1. $s \subseteq S$ and $t \subseteq T$.
2. $\forall k \in \{1, ..., |s|\}$ such that $A(i_k, j) = 1$, then $j \in \{j_1, ..., j_{|t|}\}$.
3. $\forall l \in \{1, ..., |t|\}$ such that $A(i, j_l) = 1$, then $i \in \{i_1, ..., i_{|s|}\}$.
4. $\exists k \in \{1, ..., |s|\}$ and $\exists l \in \{1, ..., |t|\}$ such that $A(i_k, j_l) = 1$.

Condition 1 guarantees that (s, t) is a phrase pair and not just a pair of strings. Condition 2 says that if a word in s is aligned to one or more words in T, then all such target words must appear in t. Condition 3 is the equivalent of Condition 2 for the target words. Condition 4 guarantees the existence of at least one word alignment in (s, t).

For a sentence pair (S, T), the set of all consistent pairs with an alignment matrix A is denoted by $P(S, T)$. Figure 1(a) shows an example of a sentence pair with an alignment matrix together with all its consistent pairs.

In SMT the extraction of each consistent pair (s, t) from (S, T) is followed by a statistic $f(s, t; S, T)$. Typically $f(s, t; S, T)$ counts the occurrences of (s, t) in (S, T). By considering all sentence pairs in the training data, the translation probability is constructed as

$$p(t|s) = \frac{\sum_{(S,T)} f(s,t;S,T)}{\sum_{(S,T)} \sum_{t'} f(s,t';S,T)}, \quad (2)$$

and similarly for $p(s|t)$. Finally, the entries of the phrase table consist of all extracted phrase pairs, their corresponding translation probabilities and other models which we do not discuss here.

3 Consistency and Components

For a given sentence pair (S, T) and a fixed word alignment matrix A, our aim is to show the equivalence between consistency and connectivity properties of the graph formed by (S, T) and A. Moreover, we explain that the way in which measurements are performed is not compatible , in principle, with the underlying structure. We start with some basic definitions from graph theory (see for example (Harary, 1969)).

Let $G = (V, E)$ be a graph with vertex set V and edge set E. Throughout this work, vertices represent words and edges represent word alignments, but the latter will be further generalized in Section 4. A *subgraph* $H = (V', E')$ of G is a graph with $V' \subseteq V$, $E' \subseteq E$ and the property that for each edge in E', both its endpoints are in V'. A *path* in G is a sequence of edges which connect a sequence of distinct vertices. Two vertices $u, v \in V$ are called connected if G contains a path from u to v. G is said to be *connected* if every pair of vertices in G is connected.

A connected component, or simply *component*, of G is a maximal connected subgraph of G. G is called *bipartite* if V can be partitioned in sets V_S and V_T, such that every edge in E connects a vertex in V_S to one in V_T. The disjoint union of graphs, or simply *union*, is an operation on graphs defined as follows. For n graphs with disjoint vertex sets $V_1, ..., V_n$ (and hence disjoint edge sets), their union is the graph $(\cup_{i=1}^n V_i, \cup_{i=1}^n E_i)$.

Consider the graph G whose vertices are the words of the source and target sentences, and whose edges are induced by the non-zero entries

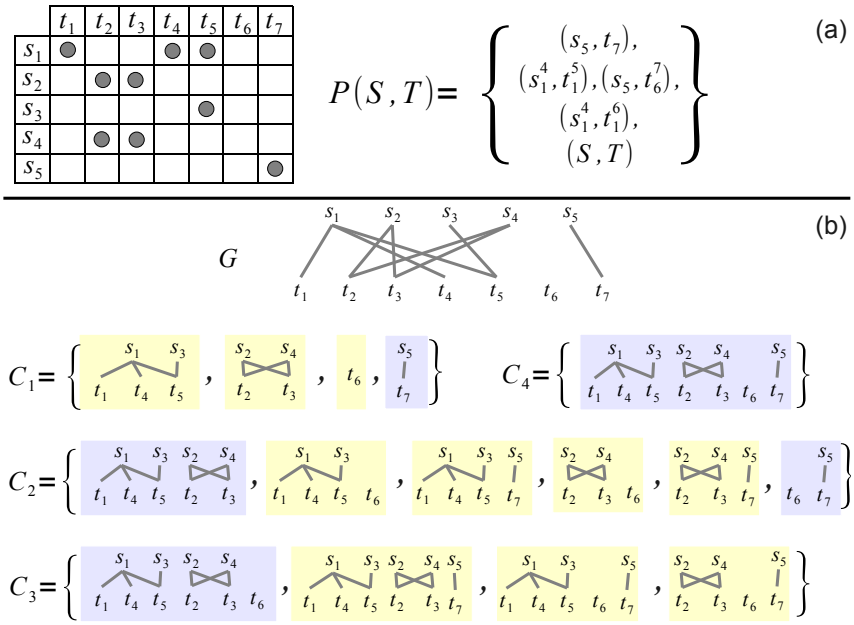

Figure 1: (a) Left: Sentence pair with an alignment matrix. Dots indicate existence of word alignments. Right: All consistent pairs. (b) The graph representation of the matrix in (a), and the sets generated by components of the graph. Dark shading indicates consistency.

of the matrix A. There are no edges between any two source-type vertices nor between any two target-type vertices. Moreover, the source and target language vocabularies are assumed to be disjoint and thus G is bipartite. The set of all components of G is defined as C_1 and let k denote its cardinality, i.e., $|C_1| = k$. From the members of C_1 we further construct sets $C_2, ..., C_k$ as follows: For each i, $2 \leq i \leq k$, any member of C_i is formed by the union of any i distinct members of C_1. In other words, any member of C_i is a graph with i components and each such component is a member of C_1. The cardinality of C_i is clearly $\binom{k}{i}$, for every i, $1 \leq i \leq k$.

Note that $C_k = \{G\}$, since G is the union of all members of C_1. Moreover, observe that $C_* = \cup_{i=1}^{k} C_i$ is the set of graphs that can be generated by all possible unions of G's components. In that sense

$$C = \{\emptyset\} \cup C_* \tag{3}$$

is the power set of G. Indeed we have $|C| = 1 + \sum_{i=1}^{k} \binom{k}{i} = 2^k$ as required.[1]

Figure 1(b) shows the graph G and the associated sets C_i of (S, T) and A in Figure 1(a). Note the bijective correspondence between consistent pairs and the phrase pairs that can be extracted from the vertices of the members of the sets C_i. This is a consequence of consistency Conditions 2 and 3, since they provide the sufficient conditions for component formation.

In general, if a pair of strings (s, t) satisfies the consistency Conditions 2 and 3, then it can be extracted from the vertices of a graph in C_i, for some i. Moreover, if Conditions 1 and 4 are also satisfied, i.e., if (s, t) is consistent, then we can write

$$P(S,T) = \bigcup_{i=1}^{k} \{\ (S_H, T_H) :\ H \in C_i, \\ S_H \subseteq S,\ T_H \subseteq T\ \}, \tag{4}$$

where S_H denotes the extracted string from the source-type vertices of H, and similarly for T_H. Having established this relationship, when referring to members of C, we henceforth mean either consistent pairs or *inconsistent* pairs. The latter are pairs (S_H, T_H) for some $H \in C$ such that at least either $S_H \not\subseteq S$ or $T_H \not\subseteq T$.

The construction above shows that phrase pairs of interest to SMT are part of a carefully constructed subclass of all possible string pairs that can be extracted from (S, T). The power set C of G gives rise to a small, possibly minimal, set

[1] Here we used the fact that for any set X with $|X| = n$, the set of all subsets of X, i.e., the power set of X, has cardinality $\sum_{i=0}^{n} \binom{n}{i} = 2^n$.

in which consistent and inconsistent pairs can be *measured*.[1] In other words, since C is (by construction) a sigma-algebra, the pair (C_1, C) is a measurable space. Furthermore, one can construct a measure space (C_1, C, f), with an appropriately chosen measure $f : C \rightarrow [0, \infty)$.

Is the occurrence-counting measure f of Section 2 a good choice? Fix an ordering for C_i, and let $C_{i,j}$ denote the jth member of C_i, for all i, $1 \leq i \leq k$. Furthermore, let $\delta(x, y) = 1$, if $x = y$ and 0, otherwise. We argue by contradiction that the occurrence-counting measure

$$f(H) = \sum_{\{H' :\ H' \in C,\ H' \text{ is consistent}\}} \delta(H, H'), \quad (5)$$

fails to form a measure space. Suppose that more than one component of G is consistent, i.e., suppose that

$$1 < \sum_{j=1}^{k} f(C_{1,j}) \leq k. \quad (6)$$

By construction of C, it is guaranteed that

$$1 = f(G) = f(C_{k,1}) = f(\cup_{j=1}^{k} C_{1,j}). \quad (7)$$

The members of C_1 are pairwise disjoint, because each of them is a component of G. Thus, since f is assumed to be a measure, sigma-additivity should be satisfied, i.e., we must have

$$f(\cup_{j=1}^{k} C_{1,j}) = \sum_{j=1}^{k} f(C_{1,j}) > 1, \quad (8)$$

which is a contradiction.

In practice, the deficiency of using eq. 5 as a statistic could possibly be explained by the fact that the so-called lexical weights are used as smoothing.

4 Consistency, Components and Segmentations

In Section 3 the only relation that was assumed among source (target) words/vertices was the order of appearance in the source (target) sentence. As a result, the graph representation G of (S, T) and A was bipartite. There are several, linguistically motivated, ways in which a general graph can be obtained from the bipartite graph G. We explain that the minimal linguistic structure, namely sentence segmentations, can provide a generalization of the construction introduced in Section 3.

Let X be a finite set of consecutive integers. A consecutive partition of X is a partition of X such that each part consists of integers consecutive in X. A segmentation σ of a source sentence S is a consecutive partition of $\{1, ..., |S|\}$. A part of σ, i.e., a segment, is intuitively interpreted as a phrase in S. In the graph representation G of (S, T) and A, a segmentation σ of S is realised by the existence of edges between consecutive source-type vertices whose labels, i.e., word positions in S, appear in the same segment of σ. The same argument holds for a target sentence and its words; a target segmentation is denoted by τ.

Clearly, there are $2^{|S|-1}$ possible ways to segment S and, given a fixed alignment matrix A, the number of all possible graphs that can be constructed is thus $2^{|S|+|T|-2}$. The bipartite graph of Section 3 is just one possible configuration, namely the one in which each segment of σ consists of exactly one word, and similarly for τ. We denote this segmentation pair by (σ_0, τ_0).

We now turn to extracting consistent pairs in this general setting from all possible segmentations (σ, τ) for a sentence pair (S, T) and a fixed alignment matrix A. As in Section 3, we construct graphs $G^{\sigma,\tau}$, associated sets $C_i^{\sigma,\tau}$, for all i, $1 \leq i \leq k^{\sigma,\tau}$, and $C^{\sigma,\tau}$, for all (σ, τ). Consistent pairs are extracted in lieu of eq. 4, i.e.,

$$P^{\sigma,\tau}(S,T) = \bigcup_{i=1}^{k^{\sigma,\tau}} \{\ (S_H,\ T_H) :\ H \in C_i^{\sigma,\tau},\ S_H \subseteq S,\ T_H \subseteq T\ \}, \quad (9)$$

and it is trivial to see that

$$\{(S,T)\} \subseteq P^{\sigma,\tau}(S,T) \subseteq P(S,T), \quad (10)$$

for all (σ, τ). Note that $P(S, T) = P^{\sigma_0,\tau_0}(S, T)$ and, depending on the details of A, it is possible for other pairs (σ, τ) to attain equality. Moreover, each consistent pair in $P(S, T)$ can be be extracted from a member of at least one $C^{\sigma,\tau}$.

We focus on the sets $C_1^{\sigma,\tau}$, i.e., the components of $G^{\sigma,\tau}$, for all (σ, τ). In particular, we are interested in the relation between $P(S, T)$ and $C_1^{\sigma,\tau}$, for all (σ, τ). Each consistent $H \in C^{\sigma_0,\tau_0}$ can be converted into a single component by appropriately forming edges between consecutive source-type vertices and/or between consecutive target-type vertices. The resulting component will evidently be a member of $C_1^{\sigma,\tau}$, for some (σ, τ). It

[1] See Appendix for definitions.

is important to note that the conversion of a consistent $H \in C^{\sigma_0,\tau_0}$ into a single component need not be unique; see Figure 2 for a counterexample. Since (a) such conversions are possible for all consistent $H \in C^{\sigma_0,\tau_0}$ and (b) $P(S,T) = P^{\sigma_0,\tau_0}(S,T)$, it can be deduced that all possible consistent pairs can be traced in the sets $C_1^{\sigma,\tau}$, for all (σ,τ). In other words, we have:

$$P(S,T) = \bigcup_{\sigma,\tau} \{\ (S_H,\ T_H) :\ H \in C_1^{\sigma,\tau},\ S_H \subseteq S,\ T_H \subseteq T\ \}. \quad (11)$$

The above equation says that by taking sentence segmentations into account, we can recover all possible consistent pairs, by inspecting only the components of the underlying graphs.

It would be interesting to investigate the relation between measure spaces $(C_1^{\sigma,\tau}, C^{\sigma,\tau}, f^{\sigma,\tau})$ and different configurations for A. We leave that for future work and focus on the advantages provided by eq. 11.

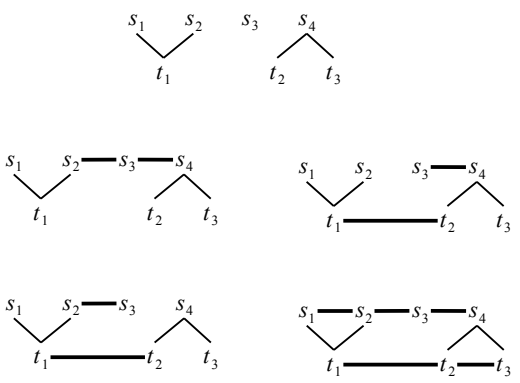

Figure 2: A graph with three components (top), and four possible conversions into a single component by forming edges between contiguous words.

5 Towards a phrase-level model that respects consistency

The aim of this section is to exploit the relation established in eq. 11 between consistent pairs and components of segmented sentence pairs. It was also shown in Section 2 that the computation of the translation models is inappropriate to describe the underlying structure. We thus suggest a phrase-based generative model in the spirit of the IBM word-based models, which is compatible with the construction of the previous sections.

5.1 Hidden variables

All definitions from the previous sections are carried over, and we introduce a new quantity that is associated with components. Let $G^{\sigma,\tau}$ and $C_1^{\sigma,\tau}$, for some (σ,τ) be as in Section 4, then the set K is defined as follows: Each member of K is a pair of (source, target) sets of segments that corresponds to the pair of (source, target) vertices of a consistent member of $C_1^{\sigma,\tau}$. In other words, K is a bisegmentation of a pair of segmented sentences that respects consistency.

Figure 3 shows three possible ways to construct consistent graphs from $(S,T) = (s_1^4, t_1^6)$, $\sigma = \{\{1,2\},\{3\},\{4\}\} \equiv \{x_1,x_2,x_3\}$ and $\tau = \{\{1\},\{2,3,4\},\{5\},\{6\}\} \equiv \{y_1,y_2,y_3,y_4\}$. In each case the exact alignment information is unknown and we have:

$$\text{(a) } K = \Big\{ \big(\{x_1\},\ \{y_1\}\big),\ \big(\{x_2\},\ \{y_2\}\big),\ \big(\{x_3\},\ \{y_3,y_4\}\big) \Big\}.$$

$$\text{(b) } K = \Big\{ \big(\{x_1,x_2\},\ \{y_1,y_2,y_3\}\big),\ \big(\{x_3\},\{y_4\}\big) \Big\}.$$

$$\text{(c) } K = \Big\{ \big(\{x_1\},\ \{y_3,y_4\}\big),\ \big(\{x_2,\ x_3\},\ \{y_1,y_2\}\big) \Big\}.$$

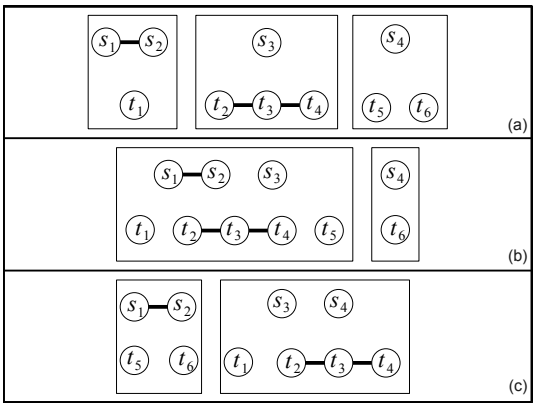

Figure 3: Three possible ways to construct consistent graphs for (s_1^4, t_1^6) and a given segmentation pair. Exact word alignment information is unknown.

In the proposed phrase-level generative model the random variables whose instances are σ, τ and

K are hidden variables. As with the IBM models, they are associated with the positions of words in a sentence, rather than the words themselves. Alignment information is implicitly identified via the consistent bisegmentation K.

Suppose we have a corpus that consists of pairs of parallel sentences (S, T), and let $f_{S,T}$ denote the occurrence count of (S, T) in the corpus. Also, let $l_S = |S|$ and $l_T = |T|$. The aim is to maximize the corpus log-likelihood function

$$\begin{aligned}\ell &= \sum_{S,T} f_{S,T} \ \log p_\theta(T|S) \\ &= \sum_{S,T} f_{S,T} \ \log \sum_{\sigma,\tau,K} p_\theta(T, \sigma, \tau, K|S), \quad (12)\end{aligned}$$

where σ, τ and K are hidden variables parameterized by a vector θ of unknown weights, whose values are to be determined. The expectation maximization algorithm (Dempster et al., 1977) suggests that an iterative application of

$$\theta_{n+1} = \arg\max_\theta \sum_{S,T} f_{S,T} \sum_{\sigma,\tau,K} p_{\theta_n}(\sigma, \tau, K|S, T) \times \log p_\theta(T, \sigma, \tau, K|S), \quad (13)$$

provides a good approximation for the maximum value of ℓ. As with the IBM models we seek probability mass functions (PMFs) of the form

$$p_\theta(T, \sigma, \tau, K|S) = p_\theta(l_T|S) p_\theta(\sigma, \tau, K|l_T, S) \times p_\theta(T|\sigma, \tau, K, l_T, S), \quad (14)$$

and decompose further as

$$p_\theta(\sigma, \tau, K|l_T, S) = p_\theta(\sigma, \tau|l_T, S) p_\theta(K|\sigma, \tau, l_T, S) \quad (15)$$

A further simplification of $p_\theta(\sigma, \tau|l_T, S) = p_\theta(\sigma|S) p_\theta(\tau|l_T)$ may not be desirable, but will help us understand the relation between θ and the PMFs. In particular, we give a formal description of $p_\theta(\sigma|S)$ and then explain that $p_\theta(K|\sigma, \tau, l_T, S)$ and $p_\theta(T|\sigma, \tau, K, l_T, S)$ can be computed in a similar way.

5.2 Constrained, biased sampling without replacement

The probability of a segmentation given a sentence can be realised in two different ways. We first provide a descriptive approach which is more intuitive, and we use the sentence $S = s_1^4$ as an example whenever necessary. The set of all possible segments of S is denoted by $seg(S)$ and trivially $|seg(S)| = |S|\big(|S| + 1\big)/2$. Each segment $x \in seg(S)$ has a nonnegative weight $\theta(x|l_S)$ such that

$$\sum_{x \in seg(S)} \theta(x|l_S) = 1. \quad (16)$$

Suppose we have an urn that consists of $|seg(S)|$ weighted balls; each ball corresponds to a segment of S. We sample without replacement with the aim of collecting enough balls to form a segmentation of S. When drawing a ball x we simultaneously remove from the urn all other balls x' such that $x \cap x' \neq \emptyset$. We stop when the urn is empty. In our example, let the urn contain 10 balls and suppose that the first draw is $\{1, 2\}$. In the next draw, we have to choose from $\{3\}$, $\{4\}$ and $\{3, 4\}$ only, since all other balls contain a '1' and/or a '2' and are thus removed. The sequence of draws that leads to a segmentation is thus a path in a decision tree. Since σ is a set, there are $|\sigma|!$ different paths that lead to its formation. The set of all possible segmentations, in all possible ways that each segmentation can be formed, is encoded by the collection of all such decision trees.

The second realisation, which is based on the notions of cliques and neighborhoods, is more constructive and will give rise to the desired PMF. A *clique* in a graph is a subset U of the vertex set such that for every two vertices $u, v \in U$, there exists an edge connecting u and v. For any vertex u in a graph, the *neighborhood* of u is defined as the set $N(u) = \{v : \{u, v\} \text{ is an edge}\}$. A *maximal clique* is a clique U that is not a subset of a larger clique: For each $u \in U$ and for each $v \in N(u)$ the set $U \cup \{v\}$ is not a clique.

Let $\mathcal{G}$ be the graph whose vertices are all segments of S and whose edges satisfy the condition that any two vertices x and x' form an edge iff $x \cap x' = \emptyset$; see Figure 4 for an example. $\mathcal{G}$ essentially provides a compact representation of the decision trees discussed above.

It is not difficult to see that a maximal clique also forms a segmentation. Moreover, the set of all maximal cliques in $\mathcal{G}$ is exactly the set of all possible segmentations for S. Thus, $p_\theta(\sigma|S)$ should satisfy

$$p_\theta(\sigma|S) = 0, \text{ if } \sigma \text{ is not a clique in } \mathcal{G}, \quad (17)$$

and

$$\sum_\sigma p_\theta(\sigma|S) = 1, \quad (18)$$

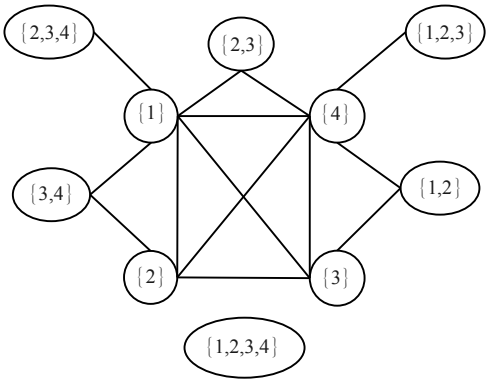

Figure 4: The graph whose vertices are the segments of s_1^4 and whose edges are formed by non-overlapping vertices.

where the sum is over all maximal cliques in $\mathcal{G}$. In our example $p_\theta(\{\ \{1\},\{1,2\}\ \}|S) = 0$, because there is no edge connecting segments $\{1\}$ and $\{1,2\}$ so they are not part of any clique.

In order to derive an explicit formula for $p_\theta(\sigma|S)$ we focus on a particular type of paths in $\mathcal{G}$. A path is called *clique-preserving*, if every vertex in the path belongs to the same clique. Our construction should be such that each clique-preserving path has positive probability of occurring, and all other paths should have probability 0. We proceed with calculating probabilities of clique-preserving paths based on the structure of $\mathcal{G}$ and the constraint of eq. 16.

The probability $p_\theta(\sigma|S)$ can be viewed as the probability of generating all clique-preserving paths on the maximal clique σ in $\mathcal{G}$. Since σ is a clique, there are $|\sigma|!$ possible paths that span its vertices. Let $\sigma = \{x_1, ..., x_{|\sigma|}\}$, and let π denote a permutation of $\{1, ..., |\sigma|\}$. We are interested in computing the probability $q_\theta(x_{\pi(1)}, ..., x_{\pi(|\sigma|)})$ of generating a clique-preserving path $x_{\pi(1)}, ..., x_{\pi(|\sigma|)}$ in $\mathcal{G}$. Thus,

$$\begin{aligned} p_\theta(\sigma|S) &= p_\theta(\{x_1, ..., x_{|\sigma|}\}|S) \\ &= \sum_\pi q_\theta(x_{\pi(1)}, ..., x_{\pi(|\sigma|)}) \\ &= \sum_\pi q_\theta(x_{\pi(1)})\ q_\theta(x_{\pi(2)}|x_{\pi(1)}) \times ... \\ &\quad ... \times q_\theta(x_{\pi(|\sigma|)}|x_{\pi(1)}, ..., x_{\pi(|\sigma|-1)}). \end{aligned} \tag{19}$$

The probabilities $q_\theta(\cdot)$ can be explicitly calculated by taking into account the following observation. A clique-preserving path on a clique σ can be realised as a sequence of vertices $x_{\pi(1)}, ..., x_{\pi(i)}, ..., x_{\pi(|\sigma|)}$ with the following constraint: If at step $i-1$ of the path we are at vertex $x_{\pi(i-1)}$, then the next vertex $x_{\pi(i)}$ should be a neighbor of all of $x_{\pi(1)}, ..., x_{\pi(i-1)}$. In other words we must have

$$x_{\pi(i)} \in N_{\pi,i} \equiv \bigcap_{l=1}^{i-1} N(x_{\pi(l)}). \tag{20}$$

Thus, the probability of choosing $x_{\pi(i)}$ as the next vertex of the path is given by

$$q_\theta(x_{\pi(i)}|x_{\pi(1)}, ..., x_{\pi(i-1)}) = \frac{\theta(x_{\pi(i)}|l_S)}{\sum_{x \in N_{\pi,i}} \theta(x|l_S)}, \tag{21}$$

if $x_{\pi(i)} \in N_{\pi,i}$ and 0, otherwise. When choosing the first vertex of the path (the root in the decision tree) we have $N_{\pi,1} = seg(S)$, which gives $q_\theta(x_{\pi(1)}) = \theta(x_{\pi(1)}|l_S)$, as required. Therefore eq. 19 can be written compactly as

$$p_\theta(\sigma|S) = \left(\prod_{i=1}^{|\sigma|} \theta(x_i|l_S)\right) \sum_\pi \frac{1}{Q_\theta(\sigma, \pi; S)}, \tag{22}$$

where

$$Q_\theta(\sigma, \pi; S) = \prod_{i=1}^{|\sigma|} \sum_{x \in N_{\pi,i}} \theta(x|l_S)\ . \tag{23}$$

The construction above can be generalized in order to derive a PMF for any random variable whose values are partitions of a set. Indeed, by allowing the vertices of $\mathcal{G}$ to be a subset of a power set, and keeping the condition of edge formation the same, probabilities of clique-preserving paths can be calculated in the same way. Figure 5 shows the graph $\mathcal{G}$ that represents all possible instances of K with $(S,T) = (s_1^4, t_1^5)$, $\sigma = \big\{\{1,2\},\{3\},\{4\}\big\}$ and $\tau = \big\{\{1\},\{2,3,4\},\{5\}\big\}$. Again each maximal clique is a possible consistent bisegmentation.

In order for this model to be complete, one should solve the maximization step of eq. 13 and calculate the posterior $p_{\theta_n}(\sigma, \tau, K|S,T)$. We are not bereft of hope, as relevant techniques have been developed (see Section 6).

6 Related Work

To our knowledge, this is the first attempt to investigate formal motivations behind the consistency method.

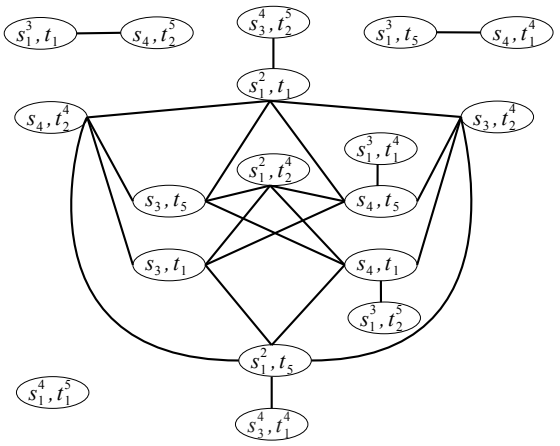

Figure 5: Similar to Figure 4 but for consistent bisegmentations with $(S,T) = (s_1^4, t_1^5)$ and a given segmentation pair (see text). For clarity, we show the phrases that are formed from joining contiguous segments in each pair, rather than the segments themselves.

Several phrase-level generative models have been proposed, almost all relying on multinomial distributions for the phrase alignments (Marcu and Wong, 2002; Zhang et al., 2003; Deng and Byrne 2005; DeNero et al., 2006; Birch et al., 2006). This is a consequence of treating alignments as functions rather than partitions.

Word alignment and phrase extraction via Inversion Transduction Grammars (Wu, 1997), is a linguistically motivated method that relies on simultaneous parsing of source and target sentences (DeNero and Klein, 2010; Cherry and Lin 2007; Neubig et al., 2012).

The partition probabilities we introduced in Section 5.2 share the same tree structure discussed in (Dennis III, 1991), which has found applications in Information Retrieval (Haffari and Teh, 2009).

7 Conclusions

We have identified the relation between consistency and components of graphs that represent word-aligned sentence pairs. We showed that phrase pairs of interest to SMT form a sigma-algebra generated by components of such graphs, but the existing occurrence-counting statistics are inadequate to describe this structure. A generalization of our construction via sentence segmentations lead to a realisation of random partitions as cases of constrained, biased sampling without replacement. As a consequence, we derived an exact formula for the probability of a segmentation of a sentence.

Appendix: Measure Space

The following standard definitions can be found in, e.g., (Feller, 1971). Let X be a set. A collection B of subsets of X is called a *sigma-algebra* if the following conditions hold:

1. $\emptyset \in B$.
2. If E is in B, then so is its complement $X \setminus E$.
3. If $\{E_i\}$ is a countable collection of sets in B, then so is their union $\cup_i E_i$.

Condition 1 guarantees that B is non-empty and Conditions 2 and 3 say that B is closed under complementation and countable unions respectively. The pair (X, B) is called a *measurable space*.

A function $f : B \to [0, \infty)$ is called a *measure* if the following conditions hold:

1. $f(\emptyset) = 0$.
2. If $\{E_i\}$ is a countable collection of pairwise disjoint sets in B, then

$$f(\cup_i E_i) = \sum_i f(E_i).$$

Condition 2 is known as *sigma-additivity*. The triple (X, B, f) is called a *measure space*.

Acknowledgments

This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430 (GALATEAS) and by the EC funded project CoSyne (FP7-ICT-4-24853).

References

Alexandra Birch, Chris Callison-Burch, Miles Osborne and Philipp Koehn. 2006. Constraining the Phrase-Based, Joint Probability Statistical Translation Model. In *Proc. of the Workshop on Statistical Machine Translation*, pages 154–157.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation. *Computational Linguistics*, vol.19(2), pages 263–312.

Colin Cherry and Dekang Lin. 2007. Inversion Transduction Grammar for Joint Phrasal Translation Modeling. In *Proc. of SSST, NAACL-HLT / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 17–24.

A.P. Dempster, N.M. Laird and D.B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, Series B (Methodological) 39(1), pages 1–38.

John DeNero, Dan Gillick, James Zhang and Dan Klein. 2006. Why Generative Phrase Models Underperform Surface Heuristics. In *Proc. of the Workshop on Statistical Machine Translation*, pages 31–38.

John DeNero and Dan Klein. 2010. Discriminative Modeling of Extraction Sets for Machine Translation. In *Proc. of the Association for Computational Linguistics (ACL)*, pages 1453–1463.

Yonggang Deng and William Byrne. 2005. HMM Word and Phrase Alignment for Statistical Machine Translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and Human Language Technology (HLT-EMNLP)*, pages 169–176.

Samuel Y. Dennis III. 1991. On the Hyper-Dirichlet Type 1 and Hyper-Liouville Distributions. *Communications in Statistics - Theory and Methods*, 20(12), pages 4069–4081.

William Feller. 1971. *An Introduction to Probability Theory and its Applications, Volume II.* John Wiley, New York.

Gholamreza Haffari and Yee Whye Teh. 2009. Hierarchical Dirichlet Trees for Information Retrieval. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL)*, pages 173–181.

Frank Harary. 1969. *Graph Theory*. Addison–Wesley, Reading, MA.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL)*, pages 48–54.

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press, Cambridge, UK.

Daniel Marcu and William Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 133–139.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori and Tatsuya Kawahara. 2012. Joint Phrase Alignment and Extraction for Statistical Machine Translation. *Journal of Information Processing*, vol. 20(2), pages 512–523.

Franz J. Och, Christoph Tillmann, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proc. of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pages 20–28.

Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23, pages 377–404.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proc. of the Association for Computational Linguistics (ACL)*, pages 523–530.

Ying Zhang, Stephan Vogel and Alex Waibel. 2003. Integrated Phrase Segmentation and Alignment Algorithm for Statistical Machine Translation. In *Proc. of the International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE).*