Translating Collocations for Bilingual Lexicons: A Statistical Approach

Frank Smadja*
NetPatrol Consulting

Kathleen R. McKeown†
Columbia University

Vasileios Hatzivassiloglou†
Columbia University

Collocations are notoriously difficult for non-native speakers to translate, primarily because they are opaque and cannot be translated on a word by word basis. We describe a program named Champollion *which, given a pair of parallel corpora in two different languages and a list of collocations in one of them, automatically produces their translations. Our goal is to provide a tool to compile bilingual lexical information above the word level in multiple languages for different domains. The algorithm we use is based on statistical methods and produces* p*-word translations of* n*-word collocations in which* n *and* p *need not be the same. For example,* Champollion *translates* make . . . decision, employment equity, *and* stock market *into* prendre . . . décision, équité en matière d'emploi, *and* bourse *respectively. Testing* Champollion *on three year's worth of the Hansards corpus yielded the French translations of 300 collocations for each year, evaluated at 73% accuracy on average. In this paper, we describe the statistical measures used, the algorithm, and the implementation of* Champollion, *presenting our results and evaluation.*

1. Introduction

Hieroglyphics remained undeciphered for centuries until the discovery of the Rosetta Stone in the beginning of the 19th century in Rosetta, Egypt. The Rosetta Stone is a tablet of black basalt containing parallel inscriptions in three different languages: Greek and two forms of ancient Egyptian writings (demotic and hieroglyphics). Jean-Francois Champollion, a linguist and egyptologist, made the assumption that these inscriptions were parallel and managed after several years of research to decipher the hieroglyphic inscriptions. He used his work on the Rosetta Stone as a basis from which to produce the first comprehensive hieroglyphics dictionary (Budge, 1989).

In this paper, we describe a modern version of a similar approach: given a large corpus in two languages, our system produces translations of common word pairs and phrases which can form the basis of a bilingual lexicon. Our focus is on the use of statistical methods for the translation of multi-word expressions such as collocations which are often idiomatic in nature. Published translations of such collocations are not readily available, even for languages such as French and English, despite the fact that collocations have been recognized as one of the main obstacles to second language acquisition (Leed and Nakhimovsky, 1979).

* The work reported in this paper was done while the author was at Columbia University. His current address is NetPatrol Consulting, Tel Maneh 6, Haifa 34363, Israel. E-mail: smadja@netvision.net.il.
† Department of Computer Science, 450 Computer Science Building, Columbia University, New York, NY 10027, USA. E-mail: kathy@cs.columbia.edu, vh@cs.columbia.edu.

© 1995 Association for Computational Linguistics

We have developed a program named *Champollion*[1] which, given a sentence-aligned parallel bilingual corpus, translates collocations (or individual words) in the source language into collocations (or individual words) in the target language. The aligned corpus is used as a reference, or *database* corpus, and represents *Champollion*'s knowledge of both languages. *Champollion* uses statistical methods to incrementally construct the collocation translation, adding one word at a time. As a correlation measure, *Champollion* uses the Dice coefficient (Dice, 1945; Sörensen, 1948) commonly used in information retrieval (Salton and McGill, 1983; Frakes and Baeza-Yates, 1992). For a given source language collocation, *Champollion* identifies individual words in the target language which are highly correlated with the source collocation thus producing a set of words. These words are then used to synthesize the translation of the source language collocation in an iterative manner. *Champollion* considers all pairs of these words and identifies any of them which are highly correlated with the source collocation. Similarly, triplets are produced by adding a highly correlated word to a highly correlated pair, and the triplets that are highly correlated with the source language collocation are passed to the next stage. This process is repeated until highly correlated combinations of words no longer can be found. *Champollion* selects the group of words with highest cardinality and correlation factor as the target collocation. Finally, it produces the correct word ordering of the target collocation by examining samples in the corpus. If word order is variable in the target collocation, *Champollion* labels it as *flexible* (as in *to take steps to* which can appear as: *took immediate steps to*, *steps were taken to*, etc.); otherwise, the correct word order is reported and the collocation is labeled *rigid*.

To evaluate *Champollion*, we used a collocation compiler, XTRACT (Smadja, 1993), to automatically produce several lists of source (English) collocations. These source collocations contain both flexible word pairs which can be separated by an arbitrary number of words, and fixed constituents such as compound noun phrases. Using XTRACT on three parts of the English data in the Hansards corpus, each representing one year's worth of data, we extracted three sets of collocations, each consisting of 300 randomly selected collocations occuring with medium frequency. We then ran *Champollion* on each of these sets, using three separate database corpora, varying in size and also taken from the Hansards corpus. We asked several people who are fluent in both French and English to judge the results, and the accuracy of *Champollion* was found to range from 65% to 78%. In our discussion of results, we show how problems for the lower score can be alleviated by increasing the size of the database corpus.

In the following sections, we first present a review of related work in statistical natural language processing dealing with bilingual data. Our algorithm depends on using a measure of correlation to find words that are highly correlated across languages. We describe the measure that we use first and then provide a detailed description of the algorithm, following this with a theoretical analysis of the performance of our algorithm. We then turn to a description of the results and evaluation. Finally, we show how the results can be used for a variety of applications, closing with a discussion of the limitations of our approach and of future work.

2. Related Work

The recent availability of large amounts of bilingual data has attracted interest in several areas, including sentence alignment (Gale and Church, 1991b; Brown, Lai, and Mercer,

1 None of the authors is affiliated with Boitet's research center on machine translation in Grenoble, France, which is also named "Champollion".

1991; Simard, Foster, and Isabelle, 1992; Gale and Church, 1993; Chen, 1993), word alignment (Gale and Church, 1991a; Brown et al., 1993; Dagan, Church, and Gale, 1993; Fung and McKeown, 1994; Fung, 1995b), alignment of groups of words (Smadja, 1992; Kupiec, 1993; van der Eijk, 1993), and statistical translation (Brown et al., 1993). Of these, aligning groups of words is most similar to the work reported here, although as we shall show, we consider a greater variety of groups than in other research. In this section, we describe work on sentence and word alignment and statistical translation, showing how these goals differ from our own, and then describe work on aligning groups of words. Note that there is additional research using statistical approaches to bilingual problems, but it is less related to ours, addressing, for example, word sense disambiguation in the source language by statistically examining context (e.g., collocations) in the source language, thus allowing appropriate word selection in the target language (Brown et al., 1991; Dagan, Itai, and Schwall, 1991; Dagan and Itai, 1994).

Our use of bilingual corpora assumes a prealigned corpus. Thus, we draw on work done at AT&T Bell Laboratories by Gale and Church (1991a; 1991b; 1993) and at IBM by Brown, Lai, and Mercer (1991) on bilingual sentence alignment. Sentence alignment programs take a paired bilingual corpus as input and determine which sentences in the target language translate which sentences in the source language. Both the AT&T and the IBM groups use purely statistical techniques based on sentence length to identify sentence pairing in corpora such as the Hansards. The AT&T group (Gale and Church, 1993) defines sentence length by the number of characters in the sentences, while the IBM group (Brown, Lai, and Mercer, 1991) defines sentence length by the number of words in the sentence. Both approaches achieve similar results and have been influential in much of the research on statistical natural language processing, including ours. It has been noted in more recent work that length based alignment programs such as these are problematic for many cases of real world parallel data, such as OCR (Optical Character Recognition) input, in which periods may not be noticeable (Church, 1993) or languages where insertions or deletions are common (Shemtov, 1993; Fung and McKeown, 1994). However, these algorithms were adequate for our purposes and could be replaced by later algorithms if needed for noisy input corpora. Sentence alignment techniques are generally used as a preprocessing stage, before the main processing component which proposes actual translations, whether of words, phrases, or full text, and they are used this way in our work as well.

Translation can be approached using statistical techniques alone. Brown et al. (1990; 1993) use a stochastic language model based on techniques used in speech recognition, combined with translation probabilities compiled on the aligned corpus, in order to do sentence translation. Their system, *Candide*, uses little linguistic and no semantic information and currently produces good quality translations for short sentences containing high frequency vocabulary, as measured by individual human evaluators (see (Berger et al., 1994) for information on recent results). While they also align groups of words across languages in the process of translation, they are careful to point out that such groups may or may not occur at constituent breaks in the sentence. In contrast, our work aims at identifying syntactically and semantically meaningful units, which may be either constituents or flexible word pairs separated by intervening words, and provides the translation of these units for use in a variety of bilingual applications. Thus, the goals of our research are somewhat different.

Kupiec (1993) describes a technique for finding noun phrase correspondences in bilingual corpora using several stages. First, as for *Champollion*, the bilingual corpus must be aligned by sentences. Then, each corpus is run through a part-of-speech tagger and noun phrase recognizer separately. Finally, noun phrases are mapped to each other using an iterative re-estimation algorithm. Evaluation was done on the 100 highest

ranking correspondences produced by the program, yielding 90% accuracy. Evaluation has not been completed for the remaining correspondences consisting of 4900 distinct English noun phrases. The author indicates that the technique has several limitations due in part to the compounded error rates of the taggers and noun phrase recognizers. Van der Eijk (1993) uses a similar approach for translating terms. His work is based on the assumption that terms are noun phrases and thus, like Kupiec, also uses sentence alignment, tagging, and a noun phrase recognizer. His work differs in the correlation measure he uses; he compares local frequency of the term (i.e., frequency in sentences containing the term) to global frequency (i.e., frequency in the full corpus), decreasing the resulting score by a weight representing the distance the target term occurs from its expected position in the corpus; this weight is small if the target term is exactly aligned with the source term and larger as the distance increases. His evaluation shows 68% precision and 64% recall. We suspect that the lower precision is due in part to the fact that van der Eijk evaluated all translations produced by the program while Kupiec only evaluated the top 2%. Note that the greatest difference between these two approaches and ours is that van der Eijk and Kupiec only handle noun phrases whereas collocations have been shown to include parts of noun phrases, categories other than noun phrases (e.g., verb phrases), as well as flexible phrases that involve words separated by an arbitrary number of other words (e.g., *to take . . . steps*, *to demonstrate . . . support*). In this work as in earlier work (Smadja, 1992), we address the full range of collocations including both flexible and rigid collocations for a variety of syntactic categories.

Another approach, begun more recently than our work, is taken by Dagan and Church (1994), who use statistical methods to translate technical terminology. Like van der Eijk and Kupiec, they pre-process their corpora by tagging and by identifying noun phrases. However, they use a word-alignment program as opposed to sentence alignment and they also include single words as candidates for technical terms. One of the major differences between their work and ours is that, like van Eijk and Kupiec, they handle translation of uninterrupted sequences of words only and thus, do not handle the broader class of flexible collocations. Their system, *Termight*, first extracts candidate technical terms, presenting them to a terminologist for filtering. Then, *Termight* identifies candidate translations for each occurrence of a source term by using the word alignment to find the first and last target positions aligned with any words of the source terms. All candidate translations for a given source term are sorted by frequency and presented to the user, along with a concordance. Because *Termight* does not use additional correlation statistics, relying instead only on the word alignment, it will find translations for infrequent terms; none of the other approaches, including *Champollion*, can make this claim. Accuracy, however, is considerably lower; the most frequent translation for a term is correct only 40% of the time (compare with *Champollion*'s 73% accuracy). Since *Termight* is fully integrated within a translator's editor (another unique feature) and is used as an aid for human translators, it gets around the problem of accuracy by presenting the sorted list of translations to the translator for a choice. In all cases, the correct translation was found in this list and translators were able to speed up both the task of identifying technical terminology and translating terms.

Other recent related work aims at using statistical techniques to produce translations of single words (Fung and McKeown, 1994; Wu and Xia, 1994; Fung, 1995b) as opposed to collocations or phrases. Wu and Xia (1994) employed an estimation-maximization technique to find the optimal word alignment from previously sentence-aligned clean parallel corpora[2], with additional significance filtering. The work by Fung and McKeown

2 These corpora had little noise. Most sentences neatly corresponded to translations in the paired corpus,

(1994) and Fung (1995b) is notable for its use of techniques suitable to Asian/Indo-European language pairs as well as Indo-European language pairs. Given that Asian languages differ considerably in structure from Romance languages, statistical methods that were previously proposed for pairs of European languages do not work well for these pairs. Fung and McKeown's work also focuses on word alignment from noisy parallel corpora where there are no clear sentence boundaries or perfect translations.

Work on the translation of single words into multi-word sequences integrating techniques for machine readable dictionaries with statistical corpus analysis (Klavans and Tzoukermann, 1990; Klavans and Tzoukermann, 1996) is also relevant. While this work focuses on a smaller set of words for translation (movement verbs), it provides a sophisticated approach using multiple knowledge sources to address both one-to-many word translations and the problem of sense disambiguation. Given only one word in the source, their system, *BICORD*, uses the corpus to extend dictionary definitions and provide translations that are appropriate for a given sense but do not occur in the dictionary, producing a bilingual lexicon of movement verbs as output.

3. Collocations and Machine Translation

Collocations, commonly occurring word pairs and phrases, are a notorious source of difficulty for non-native speakers of a language (Leed and Nakhimovsky, 1979; Benson, 1985; Benson, Benson, and Ilson, 1986). This is because they cannot be translated on a word by word basis. Instead, a speaker must be aware of the meaning of the phrase as a whole in the source language and know the common phrase typically used in the target language. While collocations are not predictable on the basis of syntactic or semantic rules, they can be observed in language and thus must be learned through repeated usage. For example, in American English one says *set the table* while in British English the phrase *lay the table* is used. These are expressions that have evolved over time. It is not the meaning of the words *lay* and *set* that determines the use of one or the other in the full phrase. Here the verb functions as a *support* verb; it derives its meaning in good part from the object in this context and not from its own semantic features. In addition, such collocations are flexible. The constraint is between the verb and its object and any number of words may occur between these two elements (e.g., *You will be setting a gorgeously decorated and lavishly appointed table designed for a King*). Collocations also include rigid groups of words that do not change from one context to another, such as compounds, as in *Canadian Charter of rights and freedoms*.

To understand the difficulties that collocations pose for translation, consider sentences (1e) and (1f) in Figure 1. Although these sentences are relatively simple, automatically translating (1e) as (1f) involves several problems. Inability to translate on a word by word basis is due in part to the presence of collocations. For example, the English collocation *to demonstrate support* is translated as *prouver son adhésion*. This translation uses words that do not correspond to individual words in the source; the English translation of *prouver* is *prove* and *son adhésion* translates as *one's adhesion*. As a phrase, however, *prouver son adhésion* carries the same meaning as the source phrase. Other groups of words in (1e) cause similar problems, including *to take steps to*, *provisions of the Charter*, and *to enforce provisions*. These groups are identified as collocations for a variety of reasons. For example, *to take steps* is a collocation because *to take* is used here as a support verb for the noun *steps*. The agent *our government* doesn't actually physically take anything; rather it has begun the process of enforcement through small, concrete actions. While

with few extraneous sentences.

(1e) "Mr. Speaker, our Government has demonstrated its support for these important principles by taking steps to enforce the provisions of the Charter more vigorously."

(1f) "Monsieur le Président, notre gouvernement a prouvé son adhésion à ces importants principes en prenant des mesures pour appliquer plus systématiquement les préceptes de la Charte."

Figure 1
Example pair of matched sentences from the Hansards corpus.

the French translation *en prenant des mesures* does use the French for *take*, the object is the translation of a word that does not appear in the source, *measures*. While these are flexible collocations and there are variations in word order, *provisions of the Charter* is a compound that is very commonly used as a whole in a much more rigid way.

This example also illustrates that collocations are domain dependent, often forming part of a sublanguage. For example, *Mr. Speaker* is the proper way to refer to the speaker in the Canadian Parliament when speaking English. The French equivalent, *Monsieur le Président*, is not the literal translation but instead uses the translation of the term *President*. While this is an appropriate translation for the Canadian Parliament, in different contexts another translation would be better. Note that these problems are quite similar to the difficulties in translating technical terminology, which also is usually part of a particular technical sublanguage (Dagan and Church, 1994). The ability to automatically acquire collocation translations is thus a definite advantage for sublanguage translation. When moving to a new domain and sublanguage, translations that are appropriate can be acquired by running *Champollion* on a new corpus from that domain.

Since in some instances parts of a sentence can be translated on a word by word basis, a translator must know when a full phrase or pair of words must be considered for translation and when a word by word technique will suffice. Two tasks must therefore be considered:

1. Identify collocations, or phrases which cannot be translated on a word by word basis, in the source language.
2. Provide adequate translation for these collocations.

For both tasks, general knowledge of the two languages is not sufficient. It is also necessary to know the expressions used in the sublanguage, since we have seen that idiomatic phrases often have different translations in a restricted sublanguage than in general usage. In order to produce a fluent translation of a full sentence, it is necessary to know the specific translation for each of the source collocations.

We use XTRACT (Smadja and McKeown, 1990; Smadja, 1991a; Smadja, 1993), a tool we previously developed, to identify collocations in the source language (task 1). XTRACT works in three stages. In the first stage, word pairs are identified that co-occur significantly often. These words can be separated by up to four intervening words and thus constitute flexible collocations. In the second stage, XTRACT identifies combinations of stage one word pairs with other words and phrases, producing compounds and idiomatic templates (i.e., phrases with one or more holes to be filled by specific syntactic types). In the final stage, XTRACT filters any pairs that do not consistently occur in the

same syntactic relation, using a parsed version of the corpus. This tool has been used in several projects at Columbia University and has been distributed to a number of research and commercial sites worldwide.

XTRACT has been developed and tested on English only input. For optimal performance, XTRACT itself relies on other tools such as a part-of-speech tagger and a robust parser. Although such tools are becoming more and more available in many languages, they are still hard to find. We have thus assumed in *Champollion* that these tools were only available in one of the two languages (English), named the source language throughout the paper.

4. The Similarity Measure

In order to rank the proposed translations so that the best one is selected, *Champollion* uses a quantitative measure of correlation between the source collocation and its complete or partial translations. This measure is also used to reduce the search space to a manageable size, by filtering out partial translations that are not highly correlated with the source collocation. In this section we discuss the properties of similarity measures that are appropriate for our application. We explain why the Dice coefficient meets these criteria and why this measure is more appropriate than another frequently used measure, mutual information.

Our approach is based on the assumption that each collocation is unambiguous in the source language and has a unique translation in the target language (at least in a clear majority of the cases). In this way, we can ignore the context of the collocations and their translations, and base our decisions only on the patterns of co-occurrence of each collocation and its candidate translations across the entire corpus. This approach is quite different from those adopted for the translation of single words (Klavans and Tzoukermann, 1990; Dorr, 1992; Klavans and Tzoukermann, 1996), as in the latter case polysemy cannot be ignored; indeed, the problem of sense disambiguation has been linked to the problem of translating ambiguous words (Brown et al., 1991; Dagan, Itai, and Schwall, 1991; Dagan and Itai, 1994). The assumption of a single meaning per collocation was based on our previous experience with English collocations (Smadja, 1993), is supported for less opaque collocations by the fact that their constituent words tend to have a single sense when they appear in the collocation (Yarowsky, 1993), and was verified during our evaluation of *Champollion* (Section 7).

We construct a mathematical model of the events we want to correlate, namely of the appearance of any word or group of words in the sentences of our corpus, as follows: To each group of words G, in either the source or the target language, we map a binary random variable X_G that takes the value "1" if G appears in a particular sentence and "0" if not. Then, the corpus of paired sentences comprising our database represents a collection of samples for the various random variables X for the various groups of words. Each new sentence in the corpus provides a new independent sample for every variable X_G. For example, if G is *unemployment rate* and the words *unemployment rate* appear only in the fifth and fifty-fifth sentences of our corpus (not necessarily in that order and perhaps with other words intervening), in our sample collection X_G takes the value "1" for the fifth and fifty-fifth sentences and "0" for all other sentences in the corpus. Furthermore, for the measurement of correlation between a word group G in the source language and another word group H in the target language, we map the paired sentences in our corpus to a collection of *paired* samples for the random variables X_G and X_H. This modeling process allows us to use correlation metrics between paired samples of random variables (X_G and X_H) to measure the correlation between word groups (G and H) across languages.

There are several ways to measure the correlation of two such random variables. One measure frequently used in information retrieval is the Dice coefficient (Dice, 1945; Sörensen, 1948; Salton and McGill, 1983; Frakes and Baeza-Yates, 1992). It is defined as

$$Dice(X, Y) = \frac{2 \cdot p(X=1, Y=1)}{p(X=1) + p(Y=1)} \quad (1)$$

where $p(X, Y)$, $p(X)$, and $p(Y)$ are the joint and marginal probability mass functions of the variables X and Y respectively. Using maximum likelihood estimates for the probabilities in the above equation, we have

$$Dice(X, Y) = \frac{2 \cdot f_{XY}}{f_X + f_Y}$$

where f_X, f_Y, and f_{XY} are the absolute frequencies of appearance of "1"s for the variables X, Y, and both X and Y together respectively.

On the other hand, in computational linguistics, information-theoretic measures such as mutual information are widely used, e.g., (Bahl et al., 1986; Church and Hanks, 1990; Church et al., 1991; Dagan, Marcus, and Markovitch, 1993; Su, Wu, and Chang, 1994). In information theory, the mutual information $I(X, Y)$ between two binary random variables X and Y is defined as

$$I(X, Y) = \sum_{x \in \{0,1\}} \sum_{y \in \{0,1\}} p(X=x, Y=y) \log \frac{p(X=x, Y=y)}{p(X=x)p(Y=y)}$$

However, in computational linguistics, the term mutual information has been used most of the time to describe only a part of the above sum, namely the term from the $X=1, Y=1$ case (unweighted by the joint probability $p(X=1, Y=1)$). In other words, this alternative measure of mutual information, to which we will refer as *specific mutual information* $SI(X, Y)$, is

$$SI(X, Y) = \log \frac{p(X=1, Y=1)}{p(X=1)p(Y=1)}$$

The quantity $I(X, Y)$ is the average of $SI(X, Y)$ taken over the four combinations of values of X and Y according to the joint probability distribution $p(X, Y)$, so sometimes the term *average mutual information* is used for $I(X, Y)$.

Average mutual information expresses the difference between the entropy (information) of one of the variables and the conditional entropy of that variable given the other variable (Cover and Thomas, 1991). Thus, average mutual information measures the reduction in the uncertainty about the value of one variable that knowledge of the value of the other variable provides, averaged over all possible values of the two variables. Equivalently, average mutual information is "the information about X contained in Y" (Papoulis, 1984, page 518) (or the information about Y contained in X). Specific mutual information represents the log-likelihood ratio of the joint probability of seeing a "1" in both variables over the probability that such an event would have if the two variables were independent, and thus provides a measure of the departure from independence. The Dice coefficient, on the other hand, combines the conditional probabilities $p(X=1|Y=1)$ and $p(Y=1|X=1)$ with equal weights in a single number. This can be

shown by replacing $p(X=1,Y=1)$ on the right side of equation (1):[3]

$$\begin{aligned} Dice(X,Y) &= \frac{2\cdot p(X=1,Y=1)}{p(X=1)+p(Y=1)} = \frac{2}{\frac{p(X=1)}{p(X=1,Y=1)}+\frac{p(Y=1)}{p(X=1,Y=1)}} = \\ &= \frac{2}{\frac{p(X=1)}{p(Y=1|X=1)p(X=1)}+\frac{p(Y=1)}{p(X=1|Y=1)p(Y=1)}} = \\ &= \frac{2}{\frac{1}{p(Y=1|X=1)}+\frac{1}{p(X=1|Y=1)}} \end{aligned}$$

As is evident from the above equation, the Dice coefficient depends only on the conditional probabilities of seeing a "1" for one of the variables after seeing a "1" for the other variable, and not on the marginal probabilities of "1"s for the two variables. In contrast, both the average and the specific mutual information depend on both the conditional and the marginal probabilities. For $SI(X,Y)$ in particular, we have

$$\begin{aligned} SI(X,Y) &= \log\frac{p(X=1,Y=1)}{p(X=1)p(Y=1)} = \log\frac{p(X=1|Y=1)p(Y=1)}{p(X=1)p(Y=1)} = \\ &= \log\frac{p(X=1|Y=1)}{p(X=1)} = \log\frac{p(Y=1|X=1)}{p(Y=1)} \end{aligned} \tag{2}$$

In order to select among the three measures, we first observe that for our application 1-1 matches (paired samples where both X and Y are 1) are significant while 0-0 matches (samples where both X and Y are 0) are not. These two types of matches correspond to the cases where either both word groups of interest appear in a pair of aligned sentences or neither word group does. Seeing the two word groups in aligned sentences (a 1-1 match) certainly contributes to their association and increases our belief that one is the translation of the other. Similarly, seeing only one of them (a 1-0 or 0-1 mismatch) decreases our belief in their association. But, given the many possible groups of words that can appear in each sentence, the fact that neither of two groups of words appears in a pair of aligned sentences does not offer any information about their similarity. Even when the word groups have been observed relatively few times (together or separately), seeing additional sentences containing none of the groups of words we are interested in should not affect our estimate of their similarity.

In other words, in our case X and Y are highly asymmetric, with a "1" value (and a 1-1 match) much more informative than a "0" value (or 0-0 match). Therefore, we should select a similarity measure that is based only on 1-1 matches and mismatches. 0-0 matches should be completely ignored, as they would otherwise dominate the similarity measure, given the overall relatively low frequency of any particular word or word group in our corpus.

3 In the remaining of this discussion, we assume that $p(X=1,Y=1)$ is not zero. This is a justified assumption for our model, since we cannot ever say that two words or word groups will not occur in the same sentence or in a sentence and its translation; such an event may well happen by chance, or because the words or word groups are parts of different syntactic constituents, even for unrelated words and word groups. The above assumption guarantees that all three measures are always well-defined; in particular, it guarantees that the marginal probabilities $p(X=1)$ and $p(Y=1)$ and the conditional probabilities $p(X=1|Y=1)$ and $p(Y=1|X=1)$ are all non-zero.

Table 1
Example values of $Dice(X, Y)$, $I(X, Y)$, and $SI(X, Y)$ after interchanging 0's and 1's.

	Original variables	Transformed variables
1-1 matches	2	92
0-0 matches	92	2
1-0 and 0-1 mismatches	6	6
Total	100	100
Dice coefficient	0.4000	0.9684
Average mutual information (bits)	0.0457	0.0457
Specific mutual information (bits)	3.0000	0.0277

The Dice coefficient satisfies the above requirement of asymmetry: Adding 0-0 matches does not change any of the absolute frequencies f_{XY}, f_X, and f_Y, and so does not affect $Dice(X, Y)$. On the other hand, average mutual information depends only on the distribution of X and Y and not on the actual values of the random variables. In fact, $I(X, Y)$ is a completely symmetric measure. If the variables X and Y are transformed so that every "1" is replaced with a "0" and vice versa, the average mutual information between X and Y remains the same. This is appropriate in the context of communications for which mutual information was originally developed (Shannon, 1948), where the ones and zeros encode two different states with no special preference for either of them. But in the context of translation, exchanging the "1"s and "0"s is equivalent to considering a word or word group present when it was absent and vice versa, thus converting all 1-1 matches to 0-0 ones and 0-0 matches to 1-1 ones. As explained above, such a change should not be considered a similarity preserving one, since 1-1 matches are much more significant than 0-0 ones.

For a concrete example, consider a corpus of 100 matched sentences, where each of the word groups associated with X and Y appears five times. Furthermore, suppose that the two groups appear twice in a pair of aligned sentences and each word group also appears three times by itself. This situation is depicted in the column labeled "original variables" in Table 1. Since each word group appears two times with the other group and three times by itself, we would normally consider the source and target groups somewhat similar but not strongly related. And indeed, the value of the Dice coefficient ($\frac{2\times 2}{5+5} = 0.4$) intuitively corresponds to that assessment of similarity.[4] Now suppose that the zeros and ones in X and Y are exchanged, so that the situation is now described by the last column of Table 1. The transformed variables now indicate that out of 100 sentences, the two word groups appear together 92 times, while each appears by itself three times and there are two sentences with none of the groups. We would consider such evidence as strongly indicating very high similarity between the two groups, and indeed the Dice coefficient of the transformed variables is now $\frac{2\times 92}{95+95} = 0.9684$. However, the average mutual information of the variables would remain the same.

Specific mutual information falls somewhere in between the Dice coefficient and average mutual information: it is not completely symmetric but neither does it ignore 0-0 matches. This measure is very sensitive to the marginal probabilities (relative frequencies) of the "1"s in the two variables, tending to give higher values as these probabilities decrease. Adding 0-0 matches lowers the relative frequencies of "1"s, and therefore *always increases* the estimate of $SI(X, Y)$. Furthermore, as the marginal probabilities of the

4 Recall that the Dice coefficient is always between 0 and 1.

two word groups become very small, $SI(X,Y)$ tends to infinity, independently of the distribution of matches (including 1-1 and 0-0 ones) and mismatches, as long as the joint probability of 1-1 matches is not zero. It can be easily verified by taking the limit of $SI(X,Y)$ for $p(X=1) \rightarrow 0$ or $p(Y=1) \rightarrow 0$ in equation (2) that this happens even if the conditional probabilities $p(X=1|Y=1)$ and $p(Y=1|X=1)$ remain constant, a fact that should indicate a constant degree of relatedness between the two variables. Neither of these problems occurs with the Dice coefficient, exactly because that measure combines the conditional probabilities of "1"s in both directions without looking at the marginal distributions of the two variables. In fact, in cases such as the examples of Table 1 where $p(X=1|Y=1) = p(Y=1|X=1)$, the Dice coefficient becomes equal to these conditional probabilities.

The dependence of $SI(X,Y)$ on the marginal probabilities of "1"s shows that using it would make rare word groups look more similar than they really are. For our example in Table 1, the specific mutual information is $SI(X,Y) = \log \frac{0.02}{0.05 \times 0.05} = \log 8 = 3$ bits for the original variables, but $SI(X',Y') = \log \frac{0.92}{0.95 \times 0.95} = \log 1.019391 = 0.027707$ bits for the transformed variables. Note, however, that the change is in the opposite direction from the appropriate one, i.e., the new variables are deemed far less similar than the old ones! This can be attributed to the fact that the number of "1"s in the original variables is far smaller.

SI(X,Y) also suffers disproportionately from estimation errors when the observed counts of "1"s are very small. While all similarity measures will be inaccurate when the data is sparse, specific mutual information can produce more misleading results since it is not bounded. This is not a problem for our application, as *Champollion* applies absolute frequency thresholds to avoid considering very rare words and word groups; but it indicates another potential problem with the use of *SI* to measure similarity.

Finally, another criterion for selecting a similarity measure is its suitability for testing for a particular outcome determined by the application. In our case, we are looking for a clear-cut test that would decide when two events are correlated. Both for mutual information and the Dice coefficient this involves comparison with a threshold that has to be determined by experimentation. Although both measures are similar in that they compare the joint probability $p(X=1, Y=1)$ with the marginal probabilities, they have different asymptotic behaviors. This was demonstrated in the previous paragraphs for the cases of small and decreasing relative frequencies. Here we examine two more cases associated with specific tests. We consider the two extreme cases, where

- The two events are perfectly independent. In this case, $p(X=x, Y=y) = p(X=x)p(Y=y)$.
- The two events are perfectly correlated in the positive direction, i.e., each word group appears every time (and only when) the other appears in the corresponding sentence. Then

$$p(X=x, Y=y) = \begin{cases} 0 & \text{if } x \neq y \\ p(X=x) = p(Y=y) & \text{if } x = y \end{cases}$$

In the first case, both average and specific mutual information are equal to 0 since $\log \frac{p(X=x,Y=y)}{p(X=x)p(Y=y)} = \log 1 = 0$ for all x and y, and are thus easily testable, whereas the Dice coefficient is equal to $\frac{2 \times (p(X=1) \times p(Y=1))}{p(X=1)+p(Y=1)}$ and is thus a function of the individual frequencies of the two word groups. In this case, the test is easier to decide using mutual information. In the second case, the results are reversed; specific mutual information is

Table 2
Dice versus specific mutual information scores for the English word *today*.
The correct translation is shown in bold.

English (X)	French (Y)	f_X	f_Y	f_{XY}	$Dice(X,Y)$	$SI(X,Y)$
today	débat aujourd	3121	143	130	0.08	5.73
	débat hui	3121	143	130	0.08	5.73
	sénat hui	3121	52	46	0.03	5.69
	aujourd hui	3121	2874	2408	0.80	5.62

equal to $\log \frac{p(X=1)}{p(X=1)^2} = -\log(p(X=1))$, and it can be shown that the average mutual information becomes equal to the entropy $H(X)$ of X (or Y). Both of these measures depend on the individual probabilities (or relative frequencies) of the word groups, whereas the Dice coefficient is equal to $\frac{2 \times p(X=1)}{p(X=1)+p(X=1)} = 1$. In this case, the test is easier to decide using the Dice coefficient. Since we are looking for a way to identify positively correlated events we must be able to easily test the second case, while testing the first case is not relevant. Specific mutual information is a good measure of independence (which it was designed to measure), but good measures of independence are not necessarily good measures of similarity.

The above arguments all support the use of the Dice coefficient over either average or specific mutual information. We have confirmed the theoretically expected behavior of the similarity measures through testing. In our early work on *Champollion* (Smadja, 1992), we used specific mutual information (*SI*) as a correlation metric. After carefully studying the errors produced, we suspected that the Dice measure would produce better results for our task, according to the arguments given above.

Consider the example given in Table 2. In the table, the second column represents candidate French word pairs for translating the single word *today*. The third column gives the frequency of the word *today* in a subset of the Hansards containing 182,584 sentences. The fourth column gives the frequency of each French word pair in the French counterpart of the same corpus, and the fifth column gives the frequency of appearance of *today* and each French word pair in matched sentences. Finally, the sixth and seventh columns give the similarity scores for *today* and each French word pair computed according to the Dice measure or specific mutual information (in bits) respectively. Of the four candidates, *aujourd hui* (shown in bold) is the only correct translation.[5] We see from the table that the specific mutual information scores fail to identify *aujourd hui* as the best candidate as it is only ranked fourth. Furthermore, the four *SI* scores are very similar, thus not clearly differentiating the results. In contrast, the Dice coefficient clearly identifies *aujourd hui* as the group of words most similar to *today*, which is what we want.

After implementing *Champollion*, we attempted to generalize these results and confirm our theoretical argumentation by performing an experiment to compare *SI* and the Dice coefficient in the context of *Champollion*. We selected a set of 45 collocations with mid-range frequency identified by XTRACT and we ran *Champollion* on them using sample training corpora (databases). For each run of *Champollion*, and for each input col-

5 Note that the correct translation is really a single word in contemporary French. *Aujourd'hui* has evolved from a collocation (*au jour d'hui*) which has become so rigid that it is now considered a single word. *Hui* can still appear on its own, but *aujourd* is not a French word, so *Champollion*'s French tokenizer erroneously considered the apostrophe character as a word separator in this case. *Champollion* will correct this error by putting *aujourd* and *hui* back together and identifying them as a rigid collocation.

Table 3
Comparison of Dice and *SI* scores on a small set of examples.

	SI Correct	SI Incorrect	Total
Dice Correct	26	10	36
Dice Incorrect	0	7	7
Total	26	17	43

location, we took the final set of candidate translations of different lengths produced by *Champollion* (with the intermediate stages driven by the Dice coefficient) and compared the results obtained using both the Dice coefficient and *SI* at the last stage for selecting the proposed translation. The 45 collocations were randomly selected from a larger set of 300 collocations so that the Dice coefficient's performance on them is representative (i.e., approximately 70% of them are translated correctly by *Champollion* when the Dice measure is used), and the correct translation is always included in the final set of candidate translations. In this way, the number of erroneous decisions made when *SI* is used at the final pass is a lower bound on the number of errors that would have been made if *SI* had also been used in the intermediate stages. We compared the results and found that out of the 45 source collocations,

- 2 were not frequent enough in the database to produce any candidate translations.
- Using the Dice coefficient, 36 were correctly translated and 7 were incorrectly translated.
- Using *SI*, 26 were correctly translated and 17 incorrectly.[6]

Table 3 summarizes these results and shows the breakdown across categories. In the table, the numbers of collocations correctly and incorrectly translated when the Dice coefficient is used are shown in the second and third rows respectively. For both cases, the second column indicates the number of these cases that were correctly translated with *SI* and the third column indicates the number of these cases that were incorrectly translated with *SI*. The last column and the last row show the total number of collocations correctly and incorrectly translated when the Dice coefficient or *SI* is used respectively. From the table we see that every time *SI* produced good results, the Dice coefficient also produced good results; there were no cases with a correct translation produced by *SI* but an incorrect one by the Dice coefficient. In addition, we see that out of the 17 incorrect results produced by *SI*, the Dice coefficient corrected 10. Although based on a few cases only, this experiment confirms that the Dice coefficient outperforms *SI* in the context of *Champollion*.

Table 4 gives concrete examples in which the Dice coefficient outperforms specific mutual information. The table has a similar format as Table 2. X represents an English collocation (*credit card* or *affirmative action*), and Y represents candidate translations in French (e.g., for the *credit cards* example: *cartes*, *cartes crédit*, *cartes crédit taux*, and *cartes crédit taux paient*). The correct translations are again shown in bold. The third and

6 In this section, incorrect translations are those judged as incorrect by the authors. We did not distinguish between errors due to XTRACT (identifying an invalid English collocation) or *Champollion* (providing a wrong translation for a valid collocation).

Table 4
Dice versus specific mutual information scores on two example English collocations. The correct translation for each source collocation is shown in bold.

English (X)	French (Y)	f_X	f_Y	f_{XY}	$Dice(X,Y)$	$SI(X,Y)$
credit cards	cartes	69	89	54	0.68	2.68
	cartes crédit	69	57	52	0.83	2.86
	cartes crédit taux	69	23	22	0.48	2.88
	cartes crédit taux paient	69	2	2	0.06	2.90
affirmative action	positive	116	89	73	0.71	2.59
	positive action	116	75	73	0.76	2.66
	positive action sociale	116	2	2	0.03	2.68

fourth columns give the independent frequencies of each word group, while the fifth column gives the number of times that both groups appear in matched sentences. The two subsequent columns give the similarity values computed according to the Dice coefficient and specific mutual information (in bits) respectively. The corpus used for these examples contained 54,944 sentences in each language. We see from Table 4 that, as for the *today* example, the *SI* scores are very close to each other and fail to select the correct candidate whereas the Dice scores cover a wider range and clearly peak for the correct translation.

In conclusion, both theoretical arguments and experimental results support the choice of the Dice coefficient over average or specific mutual information for our application.[7] Consequently, we have used the Dice coefficient as the similarity measure in *Champollion*.

5. *Champollion*: The Algorithm and the Implementation

Champollion translates single words or collocations in one language into collocations (including single word translations) in a second language using the aligned corpus as a reference database. Before running *Champollion* there are two steps that must be carried out. Source and target language sentences of the database corpus must be aligned and a list of collocations to be translated must be provided in the source language. For our experiments, we used corpora that had been aligned by Gale and Church's sentence alignment program (Gale and Church, 1991b) as our input data.[8] Since our intent in this paper is to evaluate *Champollion*, we tried not to introduce errors on the training data; for this purpose, we only kept the 1-1 alignments. Indeed, more complex sentence matches tend to have a much higher alignment error rate (Gale and Church, 1991b). By doing so, we lost an estimated 10% of the text (Brown, Lai, and Mercer, 1991), which was not a problem since we had enough data. In the future, we plan on designing more flexible

7 The choice of the Dice coefficient is not crucial; for example, using the Jaccard coefficient or any other similarity measure that is monotonically related to the Dice coefficient would be equivalent. What is important is that the selected measure satisfy the conditions of asymmetry, insensitivity to marginal word probabilities, and convenience in testing for correlation. There are many other possible measures of association, and the general points made in this section may apply to them insofar as they also exhibit the properties we discussed. For example, the normalized chi-square measure (ϕ^2) used in (Gale and Church, 1991a) shares some of the important properties of average mutual information (e.g., it is completely symmetric with respect to 1-1 and 0-0 matches).

8 We are thankful to Ken Church and the AT&T Bell Laboratories for providing us with a prealigned Hansards corpus.

Table 5
Some collocations identified by XTRACT.

Collocation	Type
Canadian Charter of Rights and Freedoms	rigid
Canadian Human Rights Commission	rigid
enforce provisions	flexible
enforce vigorously	flexible
express hope	flexible
gays and lesbians	rigid
health and safety problems	rigid
health and safety rights	rigid
health equipment	rigid
health hazards	rigid
health legislation	rigid
health services	rigid
human rights	rigid
income tax return	rigid
International Human Rights Covenants	rigid
make progress	flexible
Minister of National Health and Welfare	rigid
Nova Scotia	rigid
Smoking Control Act	rigid
take effect	flexible
take initiative	flexible
take steps	flexible
unemployment rate	rigid

techniques that would work from a loosely aligned corpus (see Section 9).

To compile collocations, we used XTRACT on the English version of the Hansards. Some of the collocations retrieved are shown in Table 5. Collocations labeled "fixed", such as *International Human Rights Covenants*, are rigid compounds. Collocations labeled "flexible" are pairs of words which can be separated by intervening words or occur in the reverse order, possibly with different inflected forms.

Given a source English collocation, *Champollion* first identifies all the sentences containing the source collocation in the database corpus. It then attempts to find all words that can be a part of the translation of the collocation, producing all words that are highly correlated with the source collocation as a whole. Once this set of words is identified, *Champollion* iteratively combines these words in groups so that each group is in turn highly correlated with the source collocation. Finally, *Champollion* produces the largest group of words which has a high correlation with the source collocation as the translation.

More precisely, for a given source collocation, *Champollion* initially identifies a set S of k words that are highly correlated with the source collocation. This operation is described in detail in Section 5.1 below. *Champollion* assumes that the target collocation is a combination of some subset of these words. Its search space at this point thus consists of the powerset $\mathcal{P}(S)$ of S containing 2^k elements. Instead of computing a correlation factor for each of the 2^k elements with the source collocation, *Champollion* searches a part of this space in an iterative manner. *Champollion* first forms all pairs of words in S, evaluates the correlation between each pair and the source collocation using the Dice coefficient, and keeps only those pairs that score above some threshold. Subsequently, it constructs the three-word elements of $\mathcal{P}(S)$ which contain one of these highly correlated pairs plus

a member of S, measures their correlation with the source collocation, and keeps the triplets that score above the threshold. This process is repeated until for some value $n \leq k$ no n-word element of $\mathcal{P}(S)$ scores above the threshold. Then *Champollion* selects the best translation among the top candidates in each group of i words, $1 \leq i \leq n-1$, and determines whether the selected translation is a single word, a flexible collocation, or a rigid collocation. If rigid, *Champollion* also reports the (fixed) order in which the words in the translation appear in the corpus.

Champollion operates in four consecutive stages. Each of these stages corresponds to one step of the algorithm, with the exception of the main iteration stage, which covers steps 2 to 4 of the algorithm. An additional preprocessing stage where indexing structures are created for fast access to the corpus is needed when the corpus database is changed. We describe the algorithm below, alternating each step with a description of what is produced using the example shown in Figure 2. This figure shows information exactly as *Champollion* produces it, with the exception of italicized comments which we provided as an English gloss explaining the various components of the output. The output includes the various potential candidate translations, of incrementally larger lengths, and the final selected translation which has the best score. Note that *Champollion* correctly determines the word order in the French translation, which is the opposite from the order of the words in the English collocation.

Stage 1 — Step 1: Initialization of the work space. Starting with a source language word group (possibly a single word) W, *Champollion* identifies all words in the target language that satisfy the following two conditions:

- The value of the Dice coefficient between the word and the source collocation W is at least T_d, where T_d is an empirically chosen threshold, and
- Appear in the target language opposite the source collocation at least T_f times, where T_f is another empirically chosen threshold.

The words that pass these tests are collected in a set S, from which the final translation will eventually be produced.

The Dice threshold T_d (currently 0.10) is the major criterion that *Champollion* uses to decide which words or partial collocations should be kept as candidates for the final translation of the source collocation. In Section 6 we explain why this incremental filtering process is necessary and we show that it does not significantly adversely affect the quality of *Champollion*'s output. To our surprise, we found that the filtering process may even *increase* the quality of the proposed translation.

The absolute frequency threshold T_f (currently 5) also helps limit the size of S, by rejecting words that appear too few times opposite the source collocation. Its most important function, however, is to remove from consideration words that appear too few times for our statistical methods to be meaningful. Applying the Dice measure (or any other statistical similarity measure) to very sparse data can produce misleading results, so we use T_f as a guide for the applicability of our method to low frequency words.

It is possible to modify the thresholds T_d and T_f according to properties of the database corpus and the collocations that are translated. Such an approach would use lower values of the thresholds, especially of T_f, for smaller corpora or less frequent collocations. In that case, a separate estimation phase is needed to automatically determine the values of the thresholds. The alternative we currently support is to allow the user to replace the default thresholds dynamically during the execution of *Champollion* with values that are more appropriate for the corpus at hand.

SOURCE COLLOCATION:
official, 492
languages, 266
The numbers indicate the frequencies of the input words in the English corpus.

NUMBER OF SENTENCES IN COMMON: 167
The two words appear together in 167 English sentences.

Champollion *now gives all the candidate final translations; that is, the best translations at each stage of the iteration process. The best single word translation is thus ⟨officielles⟩, the best pair ⟨officielles, langues⟩, the best translation with 8 words ⟨suivantes, doug, déposer, lewis, pétitions, honneur, officielles, langues⟩. The word groups are treated as sets, with no ordering. The numbers are the associated similarity score (using the Dice coefficient) for the best translation at each iteration and the number of candidate translations that passed the threshold among the word groups considered at that iteration. There are thus 11 single words that pass the thresholds at the first iteration, 35 pairs of words, etc.*
CANDIDATE TRANSLATIONS:
officielles, 0.94 out of 11
officielles langues, 0.95 out of 35
honneur officielles langues, 0.45 out of 61
déposer honneur officielles langues, 0.36 out of 71
déposer pétitions honneur officielles langues, 0.34 out of 56
déposer lewis pétitions honneur officielles langues, 0.32 out of 28
doug déposer lewis pétitions honneur officielles langues, 0.32 out of 8
suivantes doug déposer lewis pétitions honneur officielles langues, 0.20 out of 1

Champollion *then selects the optimal translation, which is the translation with the highest similarity score. In this case the result is correct.*
SELECTED TRANSLATION:
officielles langues 0.951070

An example sentence in French where the selected translation is used is also shown.
EXAMPLE SENTENCE:
Le député n' ignore pas que le gouvernement compte présenter , avant la fin de l' année , un projet de révision de la Loi sur les langues officielles .

Finally, additional information concerning word order is computed and presented. For a rigid collocation such as this one, Champollion *will print for all words in the selected translation except the first one their distance from the first word. In our example, the second word ("langues") appears in most cases one word before "officielles", to form the compound "langues officielles". Note that this information is added during post processing after the translation has been selected, and it takes very little time to compute because of the indexing. In this case, it took a few seconds to compute this information.*
WORD ORDER:
officielles
langues: selected position: -1

Figure 2
Sample output of *Champollion*.

After all words have been collected in S, the initial set of possible translations P is set equal to S, and *Champollion* proceeds with the next stage.

When given *official languages* as input (see Figure 2), this step produces a set S with the following eleven words: *suivantes*, *doug*, *déposer*, *suprématie*, *lewis*, *pétitions*, *honneur*, *programme*, *mixte*, *officielles*, and *langues*.

Stage 2 — Step 2: Scoring of possible translations. In this step, *Champollion* examines all members of the set P of possible translations. For each member x of P, *Champollion*

computes the Dice coefficient between the source language collocation W and x. If the Dice coefficient is below the threshold T_d, x is discarded from further consideration; otherwise, x is saved in a set P'.

When given *official languages* as input, the first iteration of Step 2 simply sets P' to P, the second iteration selects 35 word pairs out of the possible 110 candidates, the third iteration selects 61 world triplets, and so on until the final (ninth) iteration when none of the three elements of P passes the threshold T_d and thus P' has no elements.

Stage 2 — Step 3: Identifying the locally best translation. Once the set of surviving translations P' has been computed, *Champollion* checks if it is empty. If this is detected, there cannot be any more translations to be considered, so *Champollion* proceeds to Step 5. If P' is not empty, *Champollion* locates the translation that looks locally (among all members of P' analyzed at this iteration) the best, i.e., the translation that has the highest Dice coefficient value with the source collocation. This translation is saved in a table C of candidate final translations, along with its length in words and its similarity score. *Champollion* then continues with the next step.

The first iteration of Step 3 on our example collocation would select the word *officielles* (among the 11 words in S) as the first candidate translation with a score of 0.94. On the second iteration, the word pair ⟨*officielles*, *langues*⟩ is selected (out of 35 pairs that pass the threshold) with a score of 0.95. On the third run, the word triplet ⟨*honneur*, *officielles*, *langues*⟩, is selected (out of 61 triplets) with a score of 0.45. On the eighth iteration of Step 3, P' only has one element, the 8-tuple ⟨*suivantes*, *doug*, *déposer*, *lewis*, *pétitions*, *honneur*, *officielles*, *langues*⟩ with a score of 0.20. Finally, on the ninth iteration of Step 3, P' is empty and *Champollion* goes to Step 5.

Stage 2 — Step 4: Updating the set of possible translations. Champollion updates the set of possible translations by setting P equal to the cartesian product of the surviving translations P' and the collection of words S (i.e., those related to the source collocation).[9] Then control returns to the start of Step 2.

In our example, during the first run of Step 4, a set of word pairs would be produced, including, for example, ⟨*suivantes*, *doug*⟩, ⟨*déposer*, *lewis*⟩, ⟨*pétitions*, *honneur*⟩, and ⟨*officielles*, *langues*⟩. Some of the word triplets formed during the second iteration of Step 4 on our example starting from the word pair ⟨*officielles*, *langues*⟩ are ⟨*officielles*, *langues*, *suivantes*⟩, ⟨*officielles*, *langues*, *doug*⟩, ⟨*officielles*, *langues*, *lewis*⟩, and ⟨*officielles*, *langues*, *honneur*⟩. Similarly, on the third run of Step 4 quadruplets are formed such as ⟨*officielles*, *langues*, *suivantes*, *lewis*⟩. Finally, on the eighth iteration of this step, three 9-tuples are formed, including, for example, ⟨*suivantes*, *doug*, *déposer*, *lewis*, *pétitions*, *honneur*, *officielles*, *langues*, *mixte*⟩, and are passed on to Step 2; note that none of them will pass the Dice threshold at the next iteration of Step 2.

Stage 3 — Step 5: Identifying the globally best translation. If the list of candidate final translations C is empty, *Champollion* has failed to locate a target language translation of the source collocation and stops, reporting this. Otherwise, the entry with the highest Dice coefficient with the source collocation is selected as the translation. In the case of ties, the longer of the tied collocations is selected. Note that because of the way table C is maintained, the word group selected in this way is guaranteed to correspond to a global maximum (among all word groups considered) of the similarity measure. We are

9 Actually, a somewhat smaller subset of the cartesian product is formed, since translations with repeated words are eliminated from consideration. Such word groups very rarely correspond to a valid collocation.

planning to experiment with a more sophisticated technique for selecting the globally best translation, giving more weight to the collocation length so that longer word groups are considered better choices.

If the final translation contains just one word, *Champollion* reports the result and halts. Otherwise, the following step is executed.

Step 5 in our example selects the pair ⟨*officielles*, *langues*⟩ with a score of 0.951070 as the best candidate and continues with the following step.

Stage 4 — Step 6: Determination of word ordering. Once a multi-word translation has been selected, *Champollion* examines all the sentences containing the selected translation in order to determine whether the collocation is flexible (i.e., word order is not fixed) or rigid. This is done by looking at all the sentences containing the target language collocation and determining if the words involved are used consistently (i.e., at least in 60% of the cases) in the same order and at the same distance from one another. If the collocation is found to be rigid, the word order and the interword distances are also reported. Note that although this is done as a postprocessing stage, it does not require re-reading any sentence of the corpus since the information needed has already been precomputed.

On the *official languages* example, *Champollion* determines that *langues* appears before *officielles* with no intervening words and produces the rigid collocation *langues officielles*. See Figure 2 on page 17 for more details.

5.1 Computational and implementation features

Considering the size of the corpora that must be handled by *Champollion*, special care has been taken to minimize the number of disk accesses made during processing. We have experimented on up to two full years of the Hansards corpus which amounts to some 640,000 sentences in each language or about 220 Megabytes of uncompressed text. With corpora of this magnitude, *Champollion* takes between one and two minutes to translate a collocation, thus enabling its practical use as a bilingual lexicography tool.[10]

To achieve efficient processing of the corpus database, *Champollion* is implemented in two phases: the preparation phase and the actual translation phase. The preparation phase reads in the data base corpus and indexes it for fast future access using a commercial B-tree package (Informix, 1990). Each word in the original corpus is associated with a set of pointers to all the sentences containing it and to the positions of the word in each of these sentences. The frequency of each word (in sentences) is also computed at this stage. Thus, all necessary information from the corpus database is collected at this preprocessing phase with only one pass over the corpus file. At the translation phase, only the indices are accessed.

For the translation phase, we developed an algorithm that avoids computing the Dice coefficient for French words where the result must necessarily fall below the threshold. Using the index file on the English part of the corpus, we collect all French sentences that match the source collocation, and produce a list of all words that appear in these sentences together with their frequency (in sentences) in this subset of the French corpus. This operation takes only a few seconds to perform, and yields a list of a few thousand French words. The list also contains the local frequency of these words (i.e., frequency within this subset of the French corpus), and is sorted by this frequency in decreasing order. Now we start from the top of this list and work our way downwards until we

10 *Champollion* is currently being repackaged and will be soon made available to the community. XTRACT is already being distributed freely to educational or governmental research institutions, and can be licensed for commercial use. Both tools can be requested by contacting McKeown at Columbia University.

find a word that fails any of the following tests:

1. Its local frequency is lower than the threshold T_f.
2. Its local frequency is so low that we know it would be impossible for the Dice coefficient between this word and the source collocation to be higher than the threshold T_d.

Once a word fails one of the above tests, we are guaranteed that all subsequent words in the list (with lower local frequencies) will also fail the same test. By applying these two tests and removing all closed class words from the list, we greatly reduce the number of words that must be considered. In practice, about 90–98% of the words in the list fail to meet the two tests above, so we dramatically reduce our search space without having to perform relatively expensive operations. For the remaining words in the list, we know that they will pass the threshold but we nevertheless need to compute their Dice coefficient value, so as to select the best-ranking one-word translation of the source collocation.

The first of the above tests is rather obviously valid and easy to apply. For the second test, we compute an upper bound for the Dice coefficient between the word under consideration and the source collocation. Let X and Y stand for the source collocation and the French word under consideration respectively at some step of the loop through the word list. At this point, we know the global frequency of the source collocation (f_X) and the local frequency of the candidate translation word (f_{XY}), but we don't know the global frequency of the candidate word (f_Y). We need all these three quantities to compute the Dice coefficient, but while f_X is computed once for all Y and it is very efficient to compute f_{XY} as the set of sentences matching X is identified, it is more costly to find f_Y even if a special access structure is maintained. So, we first check whether there is any possibility for this word to correlate with the source collocation highly enough to pass the Dice threshold by assuming temporarily that the word does not appear at all outside the sentences matching the source collocation. By setting $f_Y = f_{XY}$, we can efficiently compute the Dice coefficient between X and Y under this assumption:

$$Dice_a(X,Y) = \frac{2 \cdot f_{XY}}{f_X + f_Y} = \frac{2 \cdot f_{XY}}{f_X + f_{XY}}$$

Of course, this assumption most likely won't be true. But since we know that $f_{XY} \leq f_Y$, it follows that $Dice_a(X,Y)$ is never less than the true value of the Dice coefficient between X and Y.[11] So comparing $Dice_a(X,Y)$ with the Dice threshold T_d will only filter out words that are guaranteed not to have a high enough Dice coefficient value independently of their overall frequency f_Y; thus, this is the most efficient process for this task that also guarantees correctness.[12] Another possible implementation involves representing the words as integers using hashing. Then it would be possible to compute f_Y and the Dice coefficient in linear time. Our method, in comparison, takes O($n \log n$) time to sort n candidates by their local frequency f_{XY}, but it retrieves the frequency f_Y and computes the Dice coefficient for a much smaller percentage of them.

11 And actually is a tight upper bound, realized when $f_{X=0,Y=1} = 0$.

12 Heuristic filtering of words with low local frequency may be more or less efficient, depending on the word, but a higher percentage of discarded words will come at the cost of inadvertently throwing out some valid words.

6. Analysis of *Champollion*'s Heuristic Filtering Stage

In this section, we analyze the generative capacity of our algorithm. In particular, we compare it against the obvious method of exhaustively generating and testing all possible groups of k words with k varying from 1 to some maximum length of the translation m.

Our concern is whether our algorithm will actually generate all valid translations (with final Dice coefficient above the threshold), while it is clear the exhaustive algorithm would.[13] Does the filtering process we use sometimes cause our algorithm to omit a valid translation, i.e., is there a possibility that a group of words has high similarity with the source collocation (above the threshold) and at the same time one or more of its subgroups have similarity below the threshold? In the worst case, as we show below, the answer to this question is affirmative. However, if only very few translations are missed in practice, the algorithm is indeed a good choice. In this section, we first show why the filtering we use is necessary and how it can miss valid translations, and then we present the results of Monte-Carlo simulation experiments (Rubinstein, 1981) which show that with appropriate selection of the threshold, the algorithm misses very few translations, that this rate of failure can be reduced even more by using different thresholds at each level, and that the missed translations are in general the less interesting ones, so that the rejection of some of the valid (according to the Dice coefficient) translations most likely leads to an increase of *Champollion*'s performance.

6.1 Why is filtering necessary?

The exhaustive algorithm described above suffers from two disadvantages: First, the only guaranteed correct value of m is the number of words Q in the target language that have been selected in Stage 1 as viable candidate translations and pass the Dice coefficient test as single words (see Section 5.1). This set of words typically contains 100–120 words, so enumerating all its subsets is impractical. But even if we artificially impose some lower ceiling for m, say three times the length of the source language collocation, we run into the second, and more severe, problem of the exhaustive approach. We need to consider

$$\binom{Q}{1} + \binom{Q}{2} + \cdots + \binom{Q}{m}$$

candidate translations. With typical values of $Q = 110$ and $m = 10$ which we observed in our corpus, the above formula evaluates to $5.2 \cdot 10^{13}$, or more than 50 trillion candidate translations. Clearly, we need to reduce this number to a more practical value.

Our algorithm achieves that by filtering out groups of words which do not have a high similarity with the source collocation, assuming that such groups will not be part of the translation. Let r_i ($i \geq 2$) be the fraction of proposed translations with i words which pass the threshold T_d. Let also P_i be the number of translations with i words that are examined by *Champollion*, and S_i the number of these translations that actually survive the thresholds and will be used to generate the candidate translations with $i + 1$ words. Clearly, $S_1 = P_1 = Q$, $P_2 = \binom{Q}{2}$, and $S_i = r_i \cdot P_i$ for $i \geq 2$. During the generation of the candidate translations of length $i + 1$ ($i \geq 2$), each of the S_i translations of length i can combine with $Q - i$ single words that are sufficiently correlated with the source language collocation, generating $Q - i$ possible translations of length $i + 1$, since *Champollion* does not consider translations that include repeated words. However, there are up to $i + 1$

13 In this section we refer to *missed valid translations* or *failures*, using these terms to describe candidate translations that are above the Dice threshold but are nevertheless rejected due to the non-exhaustive algorithm we use. These candidate translations are not necessarily correct translations from a performance perspective.

different ways that the same set of $i+1$ words can be generated in this manner, e.g., $\langle ABC \rangle$ can be generated by adding C to $\langle AB \rangle$, adding B to $\langle AC \rangle$, or adding A to $\langle BC \rangle$. When the set of translations of length i has been filtered, it's possible that not all of the $i+1$ ways to generate a given translation of length $i+1$ are available. In general, we have

$$S_i \cdot \frac{Q-i}{i+1} \leq P_{i+1} \leq S_i \cdot (Q-i)$$

Solving the recurrences specifying the upper and lower bounds together with the equations $S_i = r_i P_i$ for $i \geq 2$ and the boundary conditions $S_1 = P_1 = Q$ and $P_2 = \binom{Q}{2}$, we get, for the lower bound ($i \geq 3$),

$$\begin{aligned} P_i &\geq (r_{i-1} r_{i-2} \cdots r_2) \cdot P_2 \cdot \frac{(Q-2)(Q-3)\cdots(Q-i+1)}{3 \cdot 4 \cdots i} \\ &= (\prod_{j=2}^{i-1} r_j) \cdot \frac{Q(Q-1)}{2} \cdot \frac{(Q-2)(Q-3)\cdots(Q-i+1)}{3 \cdot 4 \cdots i} \\ &= (\prod_{j=2}^{i-1} r_j) \frac{Q!}{i!(Q-i)!} = (\prod_{j=2}^{i-1} r_j) \binom{Q}{i} \end{aligned}$$

and with a similar derivation, for the upper bound ($i \geq 3$),

$$P_i \leq (\prod_{j=2}^{i-1} r_j) \cdot \frac{Q!}{2(Q-i)!}$$

The sums of the bounds on the values P_i for $i = 3$ to m, plus the value $P_1 + P_2 = Q + \binom{Q}{2}$, give upper and lower bounds on the total number of candidate translations generated and examined by *Champollion*. When the r_i's are high, the actual number of candidate translations will be close to the lower bound. On the other hand, low values for the r_i's (i.e., a low threshold T_d) will result in the actual number of candidate translations being close to the upper of these bounds. To estimate the average number of candidate translations examined, we make the simplifying assumption that the decisions to reject each candidate translation with i words are made independently with constant probability r_i. Under these assumptions, the probability γ_i of generating a particular candidate translation with i words is the same for all translations with length i; the same applies to the probability λ_i that a translation with i words is included into the set of translations of length i that will generate the candidate translations of length $i+1$. Clearly, $\lambda_1 = \gamma_1 = \gamma_2 = 1$ and $\lambda_i = r_i \gamma_i$ for $i \geq 2$. In order for a particular translation with $i \geq 3$ words to be generated, at least one of its i subsets with $i-1$ words must have survived the threshold. With our assumptions, we have

$$\gamma_i = 1 - (1 - \lambda_{i-1})^i$$

From this recurrence equation and the boundary conditions given above we can compute the values of γ_i and λ_i for all i. Then the expected (average) number of candidate translations with $i \geq 3$ words examined by *Champollion* will be

$$\gamma_i \cdot \binom{Q}{i}$$

Table 6
Candidate translations examined by the exact and approximate algorithms for representative word set sizes and translation lengths.

Words	Maximum translation length	Exhaustive algorithm	*Champollion*'s algorithm		
			Best	Worst	Average
50	5	$2.37 \cdot 10^6$	2,884	14,302	13,558
	10	$1.34 \cdot 10^{10}$	2,888	15,870	15,032
75	5	$1.85 \cdot 10^7$	9,696	75,331	71,129
	10	$9.74 \cdot 10^{11}$	9,748	96,346	90,880
100	5	$7.94 \cdot 10^7$	24,820	259,873	244,950
	10	$1.94 \cdot 10^{13}$	25,127	391,895	369,070
150	5	$6.12 \cdot 10^8$	104,331	1,589,228	1,496,041
	10	$1.26 \cdot 10^{15}$	108,057	3,391,110	3,190,075

and the sum of these terms for $i = 3$ to m, plus the terms Q and $\binom{Q}{2}$, gives the total complexity of our algorithm. In Table 6 we show the number of candidate translations examined by the exhaustive algorithm and the corresponding best-, worst-, and average-case behavior of *Champollion* for several values of Q and m, using empirical estimates of the r_i's.

6.2 Effects of the filtering process

We showed above that filtering is necessary to bring the number of proposed translations down to manageable levels. For any corpus of reasonable size, we can find cases where a valid translation is missed because a part of it does not pass the threshold. Let N be the size of the corpus in terms of matched sentences. Separate the N sentences into eight categories, depending on whether each of the source collocation (X) and the partial translations (i.e., A and B) appear in it. Let the counts of these sentences be n_{ABX}, $n_{AB\bar{X}}$, $n_{A\bar{B}X}$, $\ldots$, $n_{\bar{A}\bar{B}\bar{X}}$, where the absence (presence) of a bar indicates that the corresponding term is present (absent). We can then find values of the $n_{\ldots}$'s that cause the algorithm to miss a valid translation as long as the corpus contains a modest number of sentences. This happens when one or more of the parts of the final translation appear frequently in the corpus but not together with the other parts of the source collocation. This phenomenon occurs even if we are allowed to vary the Dice thresholds at each stage of the algorithm. With our current constant Dice threshold $T_d = 0.1$, we may miss a valid translation as long as the corpus contains at least 20 sentences.

While our algorithm will necessarily miss some valid translations, this is a worst case scenario. To study the average-case behavior of our algorithm, we simulated its performance with randomly selected points with integer non-negative coordinates (n_{ABX}, $n_{AB\bar{X}}$, $n_{A\bar{B}X}$, $n_{A\bar{B}\bar{X}}$, $n_{\bar{A}BX}$, $n_{\bar{A}B\bar{X}}$, $n_{\bar{A}\bar{B}X}$) from the hyperplane defined by the equation

$$n_{ABX} + n_{AB\bar{X}} + n_{A\bar{B}X} + n_{A\bar{B}\bar{X}} + n_{\bar{A}BX} + n_{\bar{A}B\bar{X}} + n_{\bar{A}\bar{B}X} = N_0$$

where N_0 is the number of "interesting" sentences in the corpus for the translation under consideration, i.e., the number of sentences that contain at least one of X, A, or

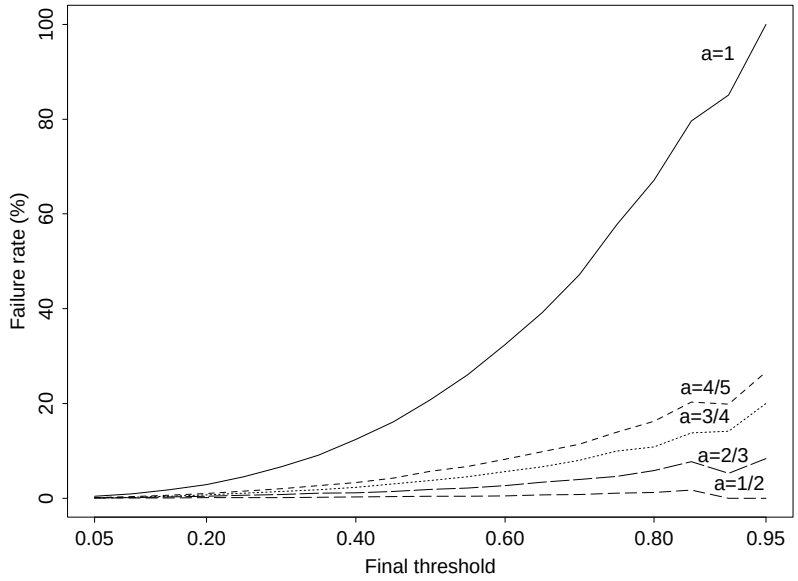

Figure 3
Failure rate of the translation algorithm with constant and increasing thresholds.
The case $\alpha = 1$ (solid line) represents the basic algorithm with no threshold changes.

B.[14] Sampling from this six-dimensional polytope in seven-dimensional space is not easy. We accomplish it by constructing a mapping from the uniform distribution to each allowed value for the $n_{\ldots}$'s, using combinatorial methods. For example, for $N_0 = 50$, there are 3,478,761 different points with $n_{ABX} = 0$ but only one with $n_{ABX} = 50$.

Using the above method, we sampled 20,000 points for each of several values for N_0 ($N_0 = 50$, 100, 500, and 1000). The results of the simulation were very similar for the different values of N_0, with no apparent pattern emerging as N_0 increased. Therefore, in the following we give averages over the values of N_0 tried.

We first measured the percentage of missed valid translations when A and/or B do not pass the threshold but AB should, for different values of the threshold parameter (solid line in Figure 3). We observed that for low values of the threshold, less than 1% of the valid translations are missed; for example, for the threshold value of 0.10 we currently use, the error rate is 0.74%. However, as the threshold increases, the rate of failure can become unacceptable.

A higher value for the threshold has two advantages: First, it offers higher selectivity, allowing less false positives (proposed translations that are not considered accurate by the human judges). Second, it speeds up the execution of the algorithm, as all fractions r_i's decrease and the overall number of candidate translations is reduced. However, as Figure 3 shows, high values of the threshold parameter cause the algorithm to miss a significant percentage of valid translations. Intuitively, we expect this problem to be alleviated if a higher threshold value is used for the final admittance of a translation, but

14 Note that the number of sentences that do not contain any of X, A, or B does not enter any of the Dice coefficients computed by *Champollion* and consequently does not affect the algorithm's decisions. As discussed in Section 4, this gives a definite advantage to the Dice method over other measures of similarity.

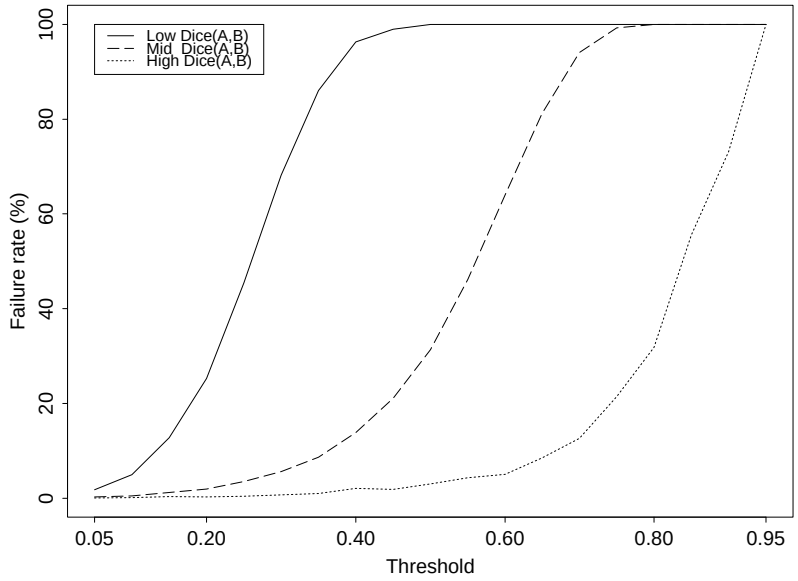

Figure 4
Failure rate of the translation algorithm according to the Dice coefficient between the partial translations.

a lower threshold is used internally when the subparts of the translation are considered. Our second simulation experiment tested this expectation for various values of the final threshold using a lower initial threshold equal to a constant $\alpha < 1$ times the final threshold. The results are represented by the remaining curves of Figure 3. Surprisingly, we found that with moderate values of α (close to 1) this method gives a very low failure rate even for high final threshold values, and is preferable to using a constant but lower threshold just to reduce the failure rate. For example, running the algorithm at an initial threshold of 0.3 and a final threshold of 0.6 gives a failure rate of 0.45%, much less than the failure rate of 6.59% which corresponds to a constant threshold of 0.3 for both stages.[15]

The above analyses show that the algorithm fails quite rarely when the threshold is low, and its performance can be improved with an increasing sequence of thresholds. We also studied cases where the algorithm does fail. For this purpose, we stratified our samples into five groups, based on the Dice coefficient between the two parts A and B (which does not directly enter the computations of the algorithm). Figure 4 shows the failure rate for the groups of low (0 to 0.2), middle (0.4 to 0.6), and high (0.8 to 1) $Dice(A, B)$ values, using the same threshold at both levels. We observe that the algorithm tends to fail much less frequently when the two parts of the final translation are strongly related. But this is desirable behavior, as a strong correlation between the two subparts

15 The curves in Figure 3 become noticeably less smooth for values of the final threshold that are greater than 0.8. This happens for all settings of α in Figure 3. This apparently different behavior for high threshold values can be traced to sampling issues. Since few of the 20,000 points in each sample meet the criterion of having $Dice(AB, X)$ greater or equal to the threshold for high final threshold values, the *estimate* of the percentage of failures is more susceptible to random variation in such cases. Furthermore, since the same sample (for a given N_0) is used for all values of α, any such random variation due to small sample size will be replicated in all curves of Figure 3.

Table 7
Failure rate of several variants of the translation algorithm for representative thresholds.

Final threshold	$\alpha = 1$	$\alpha = 3/4$	$\alpha = 1/2$	Low $Dice(A, B)$ ($\alpha = 1$)	High $Dice(A, B)$ ($\alpha = 1$)
0.05	0.39%	0.05%	0.02%	1.80%	0.02%
0.10	0.89%	0.21%	0.04%	4.99%	0.11%
0.20	2.88%	0.70%	0.13%	25.26%	0.27%
0.40	12.42%	2.29%	0.26%	96.33%	2.08%
0.80	67.11%	10.79%	1.17%	100.00%	31.83%

Table 8
Some translations produced by *Champollion*.

English collocation	French translation found by *Champollion*
additional costs	coûts supplémentaires
affirmative action	action positive
apartheid . . . South Africa	apartheid . . . afrique sud
collective agreement	convention collective
demonstrate support	prouver . . . adhésion
employment equity	équité . . . matière . . . emploi
free trade	libre-échange
freer trade	libéralisation . . . échanges
head office	siège social
health insurance	assurance-maladie
make . . . decision	prendre . . . décision
take . . . steps	prendre . . . mesures

of the word group indicates that it is indeed a collocation in the target language. The failures of the algorithm act therefore to some extent as an (unintentional) filter that rejects uninteresting translations which would have been otherwise accepted by the exhaustive method. Table 7 shows sample results from the simulation experiments, summarizing Figures 3 and 4 for several representative cases. The first column gives the threshold used at the second level, while columns 2–4 show failure rates for various values of α. For example, column 2 shows failure rates when the same threshold is used for both levels.

7. Results and Evaluation

We evaluated *Champollion* in several separate trials, varying the collocations provided as input and the database corpora used as reference. We used three different sets of collocations as input, each taken from a different year of the English half of the Hansards corpus.[16] We tested these collocations on database corpora of varying size and year, taken from the aligned Hansards. Table 8 illustrates the range of translations which *Champollion* produces. Flexible collocations are shown with a "..." indicating where additional, variable words could appear. These examples show cases where a two word collocation

16 We limited the evaluation of *Champollion* to three types of collocations, Noun-Noun (*Chamber of Commerce*), Verb-Noun (*express hope*), and Adjective-Noun (*annual deficit*) obtained using the first two stages of XTRACT.

is translated as one word (e.g., *health insurance*), a two word collocation is translated as three words (e.g., *employment equity*), and how words can be inverted in the translation (e.g., *additional costs*). In this section, we discuss the design of the separate tests and our evaluation methodology, and finally we present the results of our evaluation.

7.1 Experimental setup

We carried out three tests with *Champollion* with two database corpora and three sets of source collocations. The first database corpus (DB1) consists of 8 months of Hansards aligned data taken from 1986 (16 Megabytes, 3.5 million words) and the second database corpus consists of all of the 1986 and 1987 transcripts of the Canadian Parliament (total of around 45 Megabytes and 8.5 million words). For the first corpus (DB1), we ran XTRACT and obtained a set of around 3,000 collocations from which we randomly selected a subset of 300 for manual evaluation purposes. The 300 collocations were selected among the collocations of mid-range frequency, i.e., collocations appearing more than 10 times in the corpus. We call this first set of source collocations (C1). The second set (C2) is a set of 300 collocations similarly selected from the set of around 5,000 collocations identified by XTRACT on all data from 1987. The third set of collocations (C3) consists of 300 collocations selected from the set of around 5,000 collocations identified by XTRACT on all data from 1988. We used DB1 with both C1 (experiment 1) and C2 (experiment 2) and we used DB2 with C3 (experiment 3).

We asked three fluent bilingual speakers to evaluate the results for the different experiments. The evaluators first examined the source collocation, validating that it indeed was a word group exhibiting semantic coherence. Source collocations that seemed incorrect (i.e., mistakenly identified as collocations by XTRACT) were removed from further consideration. The evaluators then classified the translations of the remaining collocations as either correct or incorrect. In this way, we decoupled the evaluation of *Champollion* from the errors made by XTRACT.

It is clear that our classification scheme is not perfect because some cases are difficult to judge. The judges were not especially familiar with institutionalized differences of Canadian French from continental French (which is the form of French they speak). For example, without knowledge of Canadian French, it is difficult to judge if the translation of *affirmative action* is *action positive* since this term is not used in other forms of French.

One of the biggest problems for the evaluators was scoring translations of collocations with prepositions and articles. *Champollion* does not translate closed class words such as prepositions and articles. Their frequency is so high in comparison to open class words that including them in the candidate translations causes problems with the correlation metric. Evaluators generally counted a translation as incorrect if it did not contain a preposition or article when it should have. There is one exception to this general rule. When the translation should include one closed class word, it was clearly obvious what that word should be, it occurred in one empty slot in the phrase, and *Champollion* produced a rigid collocation with an empty slot at that position and with the correct open class words, they judged the translation correct. It is exactly in these cases that the closed class word could easily be filled in when examining samples in the corpus to determine word ordering. Since the collocation is rigid, the same preposition or article occurs in most cases and so it could be extracted from the samples along with word ordering. For example, when judging the translation of *assistance program* into *programme X aide* the judges knew that the missing word (X) was *d'* even without looking at the corpus. In Section 9, we describe a later version of *Champollion* in which we added the capability to identify this type of closed class words during the last stage.

Table 9
Evaluation Results.

Experiment	Invalid source collocations (XTRACT errors)	Valid source collocations		*Champollion*'s precision
		Incorrect translations	Correct translations	
C1/DB1	11%	19%	70%	78%
C2/DB1	11%	31%	58%	65%
C3/DB2	10%	23%	66%	74%
Average	11%	24%	65%	73%

7.2 Evaluation results

The results of the evaluation experiments are given in Table 9. The first column describes the experiment, the second column gives the percentage of XTRACT errors, and the next two columns give the percentages of incorrect and correct translations of source collocations in comparison to the total number of collocations. Since our previous work shows that XTRACT has accuracy of 80% (Smadja, 1991b), it is reasonable to expect a certain number of errors in the input to *Champollion*, but these should not contribute to the evaluation of *Champollion*. Consequently, we have included in the last column of the table the percentage of correct translations produced by *Champollion* in comparison to the total number of *valid* collocations supplied to it, i.e., the percentage of *Champollion*'s correct translations if XTRACT's errors are filtered from the input. This quantity is equal to the ratio of the fourth column over the sum of the third and fourth columns.[17]

The single accuracy figures shown are computed by averaging the scores of the three individual judges. However, we noted that the scores of the individual evaluators never varied by more than 2%, thus showing high agreement between the judges.[18] Such results indicate that, in general, there is a single correct answer in each case, verifying the hypothesis of a unique translation per collocation independently of context, which we postulated in Section 4. They also indicate that it is generally easy for the evaluators to identify this unique correct answer.

When the above two conditions are not met, it is prudent to guard against the introduction of (usually pro-system) bias by asking the evaluators to produce their answers independently of the system's output, as we have argued elsewhere (Hatzivassiloglou and McKeown, 1993; Hatzivassiloglou, 1996). However, for the problem at hand, the uniqueness and accessibility of the correct answer greatly alleviates the danger of introducing bias by letting the evaluators grade the translations produced by *Champollion*. Since the latter method makes more efficient use of the judges, we decided to adopt it for our evaluation.

Among the three experiments described above, our best results are obtained when the database corpus is also used as the corpus from which XTRACT identifies the source language collocations (experiment C1/DB1). In this case, not counting XTRACT errors, accuracy is rated at 78%. It should be noted that the thresholds used by *Champollion* were determined by experimenting on a separate data set and since determining the thresholds is the only training required for our statistical method, using the same corpus as the database and for extracting input collocations is not a case of testing on the training

17 The results in the table were computed with higher precision intermediate figures; rounding to integers has been applied to the figures shown.
18 It should be noted that the judges worked independently of each other without conferring.

data.

The second experiment yielded the lowest results as many input collocations simply did not appear often enough in the database corpus. However, we suspected that this could be corrected by using a larger database corpus.[19] Thus, for our third experiment, we used DB2 which contained two years of the Hansards (1986 and 1987) and drew our input collocations from yet a different year (1988). Evaluation on this third experiment raised the accuracy to nearly as high as the first experiment, yielding 74%.

8. Applications

There is a variety of potential uses for a bilingual lexicon of collocations. The most obvious are machine translation and machine assisted human translation, but other multilingual applications, including information retrieval, summarization, and computational lexicography, also require access to bilingual lexicons.

While some researchers are attempting machine translation through purely statistical techniques, the more common approach is to use some hybrid of interlingual and transfer techniques. These symbolic machine translation systems must have access to a bilingual lexicon and the ability to construct one semi-automatically would ease the development of such systems. *Champollion* is particularly promising for this purpose for two reasons. First, it constructs translations for multi-word collocations. Collocations are known to be opaque; that is, their meaning often derives from the combination of the words and not from the meaning of the individual words themselves. As a result, translation of collocations cannot be done on a word by word basis, and some representation of collocations in both languages is needed if the system is to translate fluently. Second, collocations are domain dependent. Particularly in technical domains, the collocations differ from what one finds in general use. Accordingly, the ability to automatically discover collocations for a given domain by using a new corpus as input to *Champollion* would ease the work required to transfer an MT system to a new domain.

Multilingual systems are now being developed in addition to pure machine translation systems. These systems also need access to bilingual phrases. We are currently developing a multilingual summarization system, in which we will use the results from *Champollion*. An early version of this system (McKeown and Radev, 1995) produces short summaries of multiple news articles covering the same event using as input the templates produced by information extraction systems developed under the ARPA message understanding program. Since some information extraction systems, such as General Electric's NLToolset (Jacobs and Rau, 1990), already produce similar representations for Japanese and English news articles, the addition of an English summary generator will automatically allow for English summarization of Japanese. In addition, we are planning to add a second language for the summaries. While the output is not a direct translation of input articles, collocations that appear frequently in the news articles will also appear in summaries. Thus, a list of bilingual collocations would be useful for the summarization process.

Information retrieval is another prospective application. As shown in (Maarek and Smadja, 1989) and more recently in (Broglio et al., 1995), the precision of information retrieval systems can be improved through the use of collocations in addition to the more traditional single word indexing units. This is because a collocation gives the context

19 Another factor that could affect the performance of *Champollion* in this case is that we use the same frequency and Dice thresholds independently of the size of the corpus. As we noted in Section 5, adjusting the values of these thresholds when a new database corpus is employed may result in improved performance.

in which a given word was used and thus, will help retrieve documents which use the word with the same sense and thus improve precision. The well-known *New Mexico* example in information retrieval describes an oft-encountered problem when single word searches are employed: searching for *new* and *Mexico* independently will retrieve a multitude of documents that do not relate to *New Mexico*. Automatically identifying and explicitly using collocations such as *New Mexico* at search or indexing time can help solve this problem. We have licensed XTRACT to several sites that are using it to improve the accuracy of their retrieval or text categorization systems.

A bilingual list of collocations could be used for the development of a multilingual information retrieval system. In cases where the database of texts includes documents that are written in multiple languages, the search query need only be expressed in one language. The bilingual collocations could be used to translate the query (particularly given that it may consist of collocations in addition to single words) from the input language to other languages in the database.

Another potential application, as demonstrated by Dagan and Church (1994), is machine aided human translation. For this scenario, when a translator begins work on a collocation inside a translation editor, the translation produced by *Champollion* could be provided as a prompt giving the translator the opportunity to approve it. In such cases, it may be useful to provide the top several translations produced by *Champollion* allowing the translator to choose the best as Dagan and Church do.

Finally, *Champollion* could also be used for computer assisted lexicography. Since its output includes the translation of 1 to n word phrases, *Champollion* could be used to automatically translate lexicons. While it could not translate sentences that are often used in dictionaries as examples, it could be used both for translation of individual words as well as phrases. In this way, a list of translated words could be produced automatically from a monolingual dictionary and filtered by a lexicographer.

9. Future Work

Champollion is one of the first attempts at translating lexical constructions using statistical techniques and our work has several limitations which will be addressed in future work. In this section we describe some of them and we give possible directions of research that will be investigated in the future.

Translating closed class words. In the experiments described in this paper, *Champollion* only produced partial collocations in the target language because we eliminated closed class words from our indices. There are two reasons for eliminating such words. First, they are very frequent and appear in almost any context, so that using them would blur our statistics. The second reason is one of time and space efficiency: since these words appear in many sentences in the corpus database, it is economical to remove them from the indices. However, this causes *Champollion* to produce only partial collocations, i.e., *to cause havoc* gets translated as *semer[0], désarrois[2]*. The position numbers indicate that a word is missing between the two French words. This word is the article *le* and the full collocation is *semer le désarrois*. We implemented an extension that checks the positions around the words of a rigid collocation.[20] Note that for flexible collocations the words can occur in any order, separated by any number of words, and therefore it is difficult to check whether the same preposition is consistently used. Our extension checks one word to the left and right of the collocation, plus any gaps between words. If the same

20 This extension was implemented by Ofer Wainberg, an MS student at Columbia University.

Table 10
Some translations with prepositions produced by *Champollion*.

English collocation	French translation found by *Champollion*
amount of money	somme d' argent
capital gains	gains en capital
consumer protection	la protection des consommateurs
dispute settlement mechanism	mécanisme de règlement des différends
drug abuse	l' abus des drogues
employment equity	équité en matière d'emploi
environmental protection	protection de l' environnement
federal sales tax	taxe de vente féderale

preposition is found to occur in the same position in 90% of the sentences in which the rigid collocation occurs, it is added to the output. Note that if *Champollion* were to be used as part of machine assisted human translation, another option would be to produce a ranked list of several prepositions that are used in the corpus and let the translator choose the best option.

This extension improves the fluency of the translations tremendously. For example, *employment equity* is translated as *équité en matière d' emploi* with prepositions in place of the empty slots shown in Table 8 on page 26. Table 10 shows a variety of translations produced by this extension. While we have not yet completed a full evaluation of these results, preliminary work using the evaluation of one judge only suggests that our results improve substantially.

Tools for the target language. Tools in French such as a morphological analyzer, a tagger, a list of acronyms, a robust parser, and various lists of tagged words would be most helpful and would allow us to improve our results. For example, a tagger for French would allow us to run XTRACT on the French part of the corpus, and thus to translate from either French or English as input. In addition, running XTRACT on the French part of the corpus would allow for independent confirmation of the proposed translations, which should be French collocations. Similarly, a morphological analyzer would allow us to produce richer results as several forms of the same word would be conflated. This would increase both the expected and the actual frequencies of the co-occurrence events, which has been found empirically to have a positive effect in overall performance in other problems before (Hatzivassiloglou, 1996). Note that ignoring inflectional distinctions can have sometimes a detrimental effect if only particular forms of a word participate in a given collocation. Consequently, it might be beneficial to take into account both the distribution of the base form and the differences between the distributions of the various inflected forms.

In the current implementation of *Champollion*, we were restricted to using tools for only one of the two languages, since at the time of implementation tools for French were not readily available. However, from the above discussion it is clear that certain tools would improve the system's performance.

Separating corpus-dependent translations from general ones. Champollion identifies translations for the source collocations using the aligned corpora database as its entire knowledge of the two languages. Consequently, sometimes the results are specific to the domain and seem peculiar when viewed in a more general context. For example, we already mentioned that *Mr. Speaker* was translated as *Monsieur le Président* which is obvi-

ously only valid for this domain. *Canadian family* is another example; it is often translated as *famille* (the *Canadian* qualifier is dropped in the French version). This is an important feature of the system, as in this way the sublanguage of the domain is employed for the translation. However, many of the collocations that *Champollion* identifies are general, domain-independent ones. *Champollion* cannot make any distinction between domain specific and general collocations. What is clearly needed is a way to determine the generality of each produced translation, as many translations found by *Champollion* are of general use and could be directly applied to other domains. This may be possible by intersecting the output of *Champollion* on corpora from many different domains.

Handling very low frequency collocations. The statistics we used do not produce good results when the frequencies are low. This shows clearly when comparing our evaluation results on the first two experiments. Running the collocation set C2 over the database DB1 produced our worst results, and this can be attributed to the low frequency in DB1 of many collocations in C2. Recall that C2 was extracted from a different (and larger) corpus from DB1. This problem is due not only to the frequencies of the source collocations or of the words involved but mainly to the frequencies of their "official" translations. Indeed, while most collocations exhibit unique senses in a given domain, sometimes a source collocation appearing multiple times in the corpus does not always get translated consistently into the same target collocation in the database. This sampling problem, which generally affects all statistical approaches, was not addressed in the paper. We reduced the effects of low frequencies by purposefully limiting ourselves to source collocations of frequencies higher than 10 containing individual words with frequencies higher than 15.

Analysis of the effects of our thresholds. Various thresholds are used in *Champollion*'s algorithm to reduce the search space. A threshold too low would significantly slow down the search as, according to Zipf's law (Zipf, 1949), the number of terms occurring n times in a general English corpus is a decreasing function of n^2. Unfortunately, sometimes this filtering step causes *Champollion* to miss a valid translation. For example, one of the incorrect translations made by *Champollion* is that *important factor* was translated into *facteur* (*factor*) only instead of the proper translation *facteur important*. The error is due to the fact that the French word *important* did not pass the first step of the algorithm as its Dice coefficient with *important factor* was too low. *Important* occurs a total of 858 times in the French part of the corpus and only 8 times in the right context, whereas to pass this step it would have to have appeared 10 times or more.

Although the theoretical analysis and simulation experiments of Section 6.2 show that such cases of missing the correct translation are rare, more work needs to be done in quantifying this phenomenon. In particular, experiments with actual corpus data should supplement the theoretical (based on uniform distributions) results. Furthermore, more experimentation with the values of the thresholds needs to be done, to locate the optimum trade-off point between efficiency and accuracy. An additional direction of experiments is to vary the thresholds (and especially the frequency threshold T_f) according to the size of the database corpus and the frequency of the collocation being translated.

Incorporating the length of the translation into the score. Currently our scoring method only uses the lengths of candidate translations to break a tie in the similarity measure. It seems, however, that longer translations should get a "bonus". For example, using our scoring technique the correlation of the collocation *official languages* with the French word *officielles* is equal to 0.94 and the correlation with the French collocation *langues*

officielles is 0.95. Our scoring only uses the relative frequencies of the events without taking into account that some of these events are composed of multiple single events. We plan to refine our scoring method so that the length (number of words involved) of the events is taken into account.

Using non-parallel corpora. *Champollion* requires an aligned bilingual corpus as input. However, finding bilingual corpora can be problematic in some domains. Although organizations such as the United Nations, the European Community, and governments of countries with several official languages are big producers, such corpora are still difficult to obtain for research purposes. While aligned bilingual corpora will become more available in the future, it would be helpful if we could relax the constraint for aligned data. Bilingual corpora in the same domain, which are not necessarily translations of each other, however, are more easily available. For example, news agencies such as the Associated Press and Reuters publish in several languages. News stories often relate similar facts but they are not proper translations of one another. Although they probably use equivalent terminology, totally different techniques would be necessary to be able to use such "non-alignable" corpora as databases. Ultimately, such techniques would be more useful as they would be able to extract knowledge from noisy data. While this is definitely a large research problem, our research team at Columbia University has begun work in this area (Fung and McKeown, 1994) that shows promise for noisy parallel corpora (i.e., where the target corpus may contain either additional or deleted paragraphs and where the languages themselves do not involve neat sentence by sentence translations). In addition, bilingual word correspondences extracted from non-parallel corpora with techniques such as those proposed by Fung (1995a) also look promising.

10. Conclusion

We have presented a method for translating collocations, implemented in *Champollion*. The ability to provide translations for collocations is important for three main reasons. First, because they are opaque constructions, they cannot be translated on a word by word basis. Instead, translations must be provided for the phrase as a whole. Second, collocations are domain dependent. In each domain, there exists a variety of phrases that have specific meanings, and translations that apply only in the given domain. Finally, a quick look at a bilingual dictionary even for two widely studied languages such as English and French shows that correspondences between collocations in two languages are largely unexplored. Thus, the ability to compile a set of translations for a new domain automatically will ultimately increase the portability of machine translation systems. By applying *Champollion* to a corpus in a new domain, translations for the domain specific collocations can be automatically compiled and inaccurate results filtered by a native speaker of the target language.

The output of our system is a bilingual list of collocations that can be used in a variety of multilingual applications. It is directly applicable to machine translation systems that use a transfer approach, since such systems rely on correspondences between words and phrases of the source and target languages. For interlingua systems, identification of collocations and their translations provide a means for augmenting the interlingua. Since such phrases cannot be translated compositionally, they indicate where concepts representing such phrases must be added to the interlingua. Such bilingual phrases are also useful for other multilingual tasks, including information retrieval of multilingual documents given a phrase in one language, summarization in one language of texts in another, and multilingual generation.

Finally, we have carried out three evaluations of the system on three separate years of

the Hansards corpus. These evaluations indicate that *Champollion* has high accuracy; in the best case, 78% of the French translations of valid English collocations were judged to be good. This is a good score in comparison with evaluations carried out on full machine translation systems. We conjecture that by using statistical techniques to translate a particular type of construction which is known to be easily observable in language, we can achieve better results than by applying the same technique to all constructions uniformly.

Our work is part of a paradigm of research that focuses on the development of tools using statistical analysis of text corpora. This line of research aims at producing tools which satisfactorily handle relatively simple tasks. These tools can then be used by other systems to address more complex tasks. For example, previous work has addressed low level tasks such as tagging a free-style corpus with part-of-speech information (Church, 1988), aligning a bilingual corpus (Gale and Church, 1991b; Brown, Lai, and Mercer, 1991), and producing a list of collocations (Smadja, 1993). While each of these tools is based on simple statistics and tackles elementary tasks, we have demonstrated with our work on *Champollion* that by combining them, one can reach new levels of complexity in the automatic treatment of natural languages.

Acknowledgments

This work was supported jointly by the Advanced Research Projects Agency and the Office of Naval Research under grant N00014-89-J-1782, by the Office of Naval Research under grant N00014-95-1-0745, by the National Science Foundation under grant GER-90-24069, and by the New York State Science and Technology Foundation under grants NYSSTF-CAT(91)-053 and NYSSTF-CAT(94)-013. We wish to thank Pascale Fung and Dragomir Radev for serving as evaluators, Thanasis Tsantilas for discussions relating to the average-case complexity of *Champollion*, and the anonymous reviewers for providing useful comments on an earlier version of the paper. We also thank Ofer Wainberg for his excellent work on improving the efficiency of *Champollion* and for adding the preposition extension, and Ken Church and AT&T Bell Laboratories for providing us with a prealigned Hansards corpus.

References

Bahl, Lalit R.; Brown, Peter F.; de Souza, Peter V.; and Mercer, Robert L. (1986). "Maximum Mutual Information of Hidden Markov Model Parameters for Speech Recognition." In *Proceedings, International Conference on Acoustics, Speech, and Signal Processing (ICASSP-86)*, volume 1, pages 49–52, Tokyo, Japan, April 1986. IEEE Acoustics, Speech and Signal Processing Society, Institute of Electronics and Communication Engineers of Japan, and Acoustical Society of Japan.

Benson, Morton (1985). "Collocations and Idioms." In *Dictionaries, Lexicography, and Language Learning*, edited by Robert Ilson. Pergamon Institute of English, Oxford, Great Britain, pages 61–68.

Benson, Morton; Benson, Evelyn; and Ilson, Robert (1986). *The BBI Combinatory Dictionary of English: A Guide to Word Combinations*. John Benjamins, Amsterdam and Philadelphia.

Berger, Adam L.; Brown, Peter F.; della Pietra, Stephen A.; della Pietra, Vincent J.; Gillet, John R.; Lafferty, John D.; Mercer, Robert L.; Printz, Harry; and Ureš, Luboš (1994). "The Candide System for Machine Translation." In *Proceedings, ARPA Workshop on Human Language Technology*, pages 157–162, Plainsboro, New Jersey, March 1994. ARPA Software and Intelligent Systems Technology Office, Morgan Kaufmann, San Francisco, California.

Broglio, John; Callan, James P.; Croft, W. Bruce; and Nachbar, Daniel W. (1995). "Document Retrieval and Routing Using the INQUERY System." In *Proceedings, Third Text Retrieval Conference (TREC-3)*, pages 29–39, Gaithersburg, Maryland, April 1995. National Institute of Standards and Technology (NIST).

Brown, Peter F.; Cocke, John; della Pietra, Stephen A.; della Pietra, Vincent J.; Jelinek, Fredrick; Lafferty, John D.; Mercer, Robert L.; and Roosin, Paul S. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, June 1990.

Brown, Peter F.; Lai, Jennifer C.; and Mercer, Robert L. (1991). "Aligning Sentences in

Parallel Corpora." In *Proceedings, 29th Annual Meeting of the ACL*, pages 169–184, Berkeley, California, June 1991. Association for Computational Linguistics.

Brown, Peter F.; della Pietra, Stephen A.; della Pietra, Vincent J.; and Mercer, Robert L. (1991). "Word-Sense Disambiguation Using Statistical Methods." In *Proceedings, 29th Annual Meeting of the ACL*, pages 264–270, Berkeley, California, June 1991. Association for Computational Linguistics.

Brown, Peter F.; della Pietra, Stephen A.; della Pietra, Vincent J.; and Mercer, Robert L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June 1993.

Budge, E. A. Wallis (1989). *The Rosetta Stone*. Dover Publications, New York. Originally published as *The Rosetta Stone in the British Museum*, Religious Tract Society, London, 1929.

Chen, Stanley F. (1993). "Aligning Sentences in Bilingual Corpora Using Lexical Information." In *Proceedings, 31st Annual Meeting of the ACL*, pages 9–16, Columbus, Ohio, June 1993. Association for Computational Linguistics.

Church, Kenneth W. (1988). "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text." In *Proceedings, Second Conference on Applied Natural Language Processing (ANLP-88)*, pages 136–143, Austin, Texas, February 1988. Association for Computational Linguistics.

Church, Kenneth W. (1993). "Char_align: A Program for Aligning Parallel Texts at the Character Level." In *Proceedings, 31st Annual Meeting of the ACL*, pages 1–8, Columbus, Ohio, June 1993. Association for Computational Linguistics.

Church, Kenneth W.; Gale, William A.; Hanks, Patrick; and Hindle, Donald (1991). "Using Statistics in Lexical Analysis." In *Lexical Acquisition: Using On-line Resources to Build a Lexicon*, edited by Uri Žernik. Lawrence Erlbaum, Hillsdale, New Jersey, pages 115–165.

Church, Kenneth W. and Hanks, Patrick (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1):22–29, March 1990.

Cover, Thomas M. and Thomas, Joy A. (1991). *Elements of Information Theory*. Wiley, New York.

Dagan, Ido and Church, Kenneth W. (1994). "*Termight*: Identifying and Translating Technical Terminology." In *Proceedings, Fourth Conference on Applied Natural Language Processing (ANLP-94)*, pages 34–40, Stuttgart, Germany, October 1994. Association for Computational Linguistics.

Dagan, Ido; Church, Kenneth W.; and Gale, William A. (1993). "Robust Bilingual Word Alignment for Machine-Aided Translation." In *Proceedings, Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pages 1–8, Columbus, Ohio, June 1993. Association for Computational Linguistics.

Dagan, Ido and Itai, Alon (1994). Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics*, 20(4):563–596, December 1994.

Dagan, Ido; Itai, Alon; and Schwall, Ulrike (1991). "Two Languages Are More Informative Than One." In *Proceedings, 29th Annual Meeting of the ACL*, pages 130–137, Berkeley, California, June 1991. Association for Computational Linguistics.

Dagan, Ido; Marcus, Shaul; and Markovitch, Shaul (1993). "Contextual Word Similarity and Estimation from Sparse Data." In *Proceedings, 31st Annual Meeting of the ACL*, pages 164–171, Columbus, Ohio, June 1993. Association for Computational Linguistics.

Dice, Lee R. (1945). Measures of the Amount of Ecologic Association between Species. *Journal of Ecology*, 26:297–302.

Dorr, Bonnie J. (1992). The Use of Lexical Semantics in Interlingual Machine Translation. *Machine Translation*, 7(3):135–193.

van der Eijk, Pim (1993). "Automating the Acquisition of Bilingual Terminology." In *Proceedings, Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–119, Utrecht, The Netherlands, April 1993. Association for Computational Linguistics.

Frakes, William B. and Baeza-Yates, Ricardo, editors (1992). *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey.

Fung, Pascale (1995a). "Compiling Bilingual Lexicon Entries from a Non-Parallel English-Chinese Corpus." In *Proceedings, Third Annual Workshop on Very Large Corpora*, pages 173–183, Boston, Massachusetts, June 1995.

Fung, Pascale (1995b). "A Pattern Matching Method for Finding Noun and Proper Noun Translations from Noisy Parallel Corpora." In *Proceedings, 33rd Annual Meeting of the ACL*, pages 236–243, Boston, Massachusetts, June 1995. Association for Computational Linguistics.

Fung, Pascale and McKeown, Kathleen R. (1994). "Aligning Noisy Parallel Corpora Across Language Groups: Word Pair Feature Matching by Dynamic Time Warping." In *Proceedings, First Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 81–88, Columbia, Maryland, October 1994.

Gale, William A. and Church, Kenneth W. (1991a). "Identifying Word Correspondences in Parallel Texts." In *Proceedings, DARPA Speech and Natural Language Workshop*, pages 152–157, Pacific Grove, California, February 1991. Morgan Kaufmann, San Mateo, California.

Gale, William A. and Church, Kenneth W. (1991b). "A Program for Aligning Sentences in Bilingual Corpora." In *Proceedings, 29th Annual Meeting of the ACL*, pages 177–184, Berkeley, California, June 1991. Association for Computational Linguistics.

Gale, William A. and Church, Kenneth W. (1993). A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19(1):75–102, March 1993.

Hatzivassiloglou, Vasileios (1996). "Do We Need Linguistics When We Have Statistics? A Comparative Analysis of the Contributions of Linguistic Cues to a Statistical Word Grouping System." In *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, edited by Judith L. Klavans and Philip Resnik. MIT Press, Cambridge, Massachusetts. In press.

Hatzivassiloglou, Vasileios and McKeown, Kathleen R. (1993). "Towards the Automatic Identification of Adjectival Scales: Clustering Adjectives According to Meaning." In *Proceedings, 31st Annual Meeting of the ACL*, pages 172–182, Columbus, Ohio, June 1993. Association for Computational Linguistics.

Informix (1990). *C-ISAM Programmer's Manual*. Informix Software, Inc., Menlo Park, California.

Jacobs, Paul S. and Rau, Lisa F. (1990). "The GE NLToolset: A Software Foundation for Intelligent Text Processing." In *Proceedings, 13th International Conference on Computational Linguistics (COLING-90)*, edited by Hans Karlgren, volume 3, pages 373–377, Helsinki, Finland.

Klavans, Judith L. and Tzoukermann, Evelyne (1990). "The BICORD System: Combining Lexical Information from Bilingual Corpora and Machine Readable Dictionaries." In *Proceedings, 13th International Conference on Computational Linguistics (COLING-90)*, edited by Hans Karlgren, volume 3, pages 174–179, Helsinki, Finland.

Klavans, Judith L. and Tzoukermann, Evelyne (1996). Combining Corpus and Machine-Readable Dictionary Data for Building Bilingual Lexicons. *Machine Translation*, 10(2). In press.

Kupiec, Julian M. (1993). "An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora." In *Proceedings, 31st Annual Meeting of the ACL*, pages 17–22, Columbus, Ohio, June 1993. Association for Computational Linguistics.

Leed, Richard L. and Nakhimovsky, Alexander D. (1979). Lexical Functions and Language Learning. *Slavic and East European Journal*, 23(1):104–113, Spring 1979.

Maarek, Yoelle and Smadja, Frank (1989). "Full Text Indexing Based on Lexical Relations. An Application: Software Libraries." In *Proceedings, 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, edited by Nicholas J. Belkin and C. J. van Rijsbergen, pages 198–206, Cambridge, Massachusetts, June 1989.

McKeown, Kathleen R. and Radev, Dragomir (1995). "Generating Summaries of Multiple News Articles." In *Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, edited by Edward A. Fox, Peter Ingwersen, and Raya Fidel, pages 74–82, Seattle, Washington, July 1995.

Papoulis, Athanasios (1984). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 2nd edition.

Rubinstein, Reuven Y. (1981). *Simulation and the Monte Carlo Method*. Wiley, New York.

Salton, Gerard and McGill, Michael J. (1983). *Introduction to Modern Information Retrieval*. McGraw Hill, New York.

Shannon, Claude E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

Shemtov, Hadar (1993). "Text Alignment in a Tool for Translating Revised Documents." In *Proceedings, Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 449–453, Utrecht, The Netherlands, April 1993. Association for Computational Linguistics.

Simard, Michel; Foster, George F.; and Isabelle, Pierre (1992). "Using Cognates to Align Sentences in Bilingual Corpora." In *Proceedings, Fourth International Conference*

on Theoretical and Methodological Issues in Machine Translation (TMI-92), pages 67–81, Montréal, Canada, June 1992.

Smadja, Frank (1991a). *Extracting Collocations from Text. An Application: Language Generation*. Doctoral dissertation, Department of Computer Science, Columbia University, New York, June 1991.

Smadja, Frank (1991b). "From N-grams to Collocations: An Evaluation of Xtract." In *Proceedings, 29th Annual Meeting of the ACL*, pages 279–284, Berkeley, California, June 1991. Association for Computational Linguistics.

Smadja, Frank (1992). "How to Compile a Bilingual Collocational Lexicon Automatically." In *Proceedings, AAAI-92 Workshop on Statistically-Based NLP Techniques*, pages 65–71, San Jose, California, July 1992. American Association for Artificial Intelligence.

Smadja, Frank (1993). Retrieving Collocations From Text: Xtract. *Computational Linguistics*, 19(1):143–177, March 1993.

Smadja, Frank and McKeown, Kathleen R. (1990). "Automatically Extracting and Representing Collocations for Language Generation." In *Proceedings, 28th Annual Meeting of the ACL*, pages 252–259, Pittsburgh, Pennsylvania, June 1990. Association for Computational Linguistics.

Sörensen, Thorvald J. (1948). A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analysis of the Vegetation of Danish Commons. *Biologiske Skrifter*, 5(4):1–34.

Su, Keh-Yih; Wu, Ming-Wen; and Chang, Jing-Shin (1994). "A Corpus-based Approach to Automatic Compound Extraction." In *Proceedings, 32nd Annual Meeting of the ACL*, pages 242–247, Las Cruces, New Mexico, June 1994. Association for Computational Linguistics.

Wu, Dekai and Xia, Xuanyuin (1994). "Learning an English-Chinese Lexicon from a Parallel Corpus." In *Proceedings, First Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 206–213, Columbia, Maryland, October 1994.

Yarowsky, David (1993). "One Sense Per Collocation." In *Proceedings, ARPA Workshop on Human Language Technology*, pages 266–271, Plainsboro, New Jersey, March 1993. ARPA Software and Intelligent Systems Technology Office, Morgan Kaufmann, San Francisco, California.

Zipf, George K. (1949). *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, Reading, Massachusetts.