HLTCOE Technical Reports

No. 2

Semantically Informed Machine Translation

Final Report of the 2009 Summer Camp for Advanced Language Exploration (SCALE)

SCALE 2010 Team:

Kathy Baker, Dept. of Defense
Steven Bethard, Univ. of Colorado
Michael Bloodgood, HLTCOE
Ralf Brown, Carnegie-Mellon Univ.
Chris Callison-Burch, JHU
Glen Coppersmith, HLTCOE
Bonnie Dorr, HLTCOE / Univ. of MD
Wes Filardo, HLTCOE
Kendall Giles, Va. Commonwealth Univ.
Ann Irvine, HLTCOE
Mike Kayser, BBN
Lori Levin, Carnegie-Mellon
Justin Martineau, UMBC
Jim Mayfield, HLTCOE / APL
Scott Miller, BBN
Aaron Phillips, Carnegie-Mellon
Andrew Philpot, USC/ISI
Christine Piatko, HLTCOE / APL
Lane Schwartz, Univ. of Minnesota
David Zajic, Univ. of MD

Human Language Technology Center of Excellence
810 Wyman Park Drive
Baltimore, Maryland 21211
www.hltcoe.org

HLTCOE Technical Report
No. 2

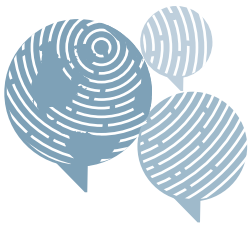
Kathy Baker, Dept. of Defense
Steven Bethard, Univ. of Colorado
Michael Bloodgood, HLTCOE
Ralf Brown, Carnegie-Mellon Univ.
Chris Callison-Burch, JHU
Glen Coppersmith, HLTCOE
Bonnie Dorr, HLTCOE / Univ. of MD
Wes Filardo, HLTCOE
Kendall Giles, Va. Commonwealth Univ.
Ann Irvine, HLTCOE
Mike Kayser, BBN
Lori Levin, Carnegie-Mellon
Justin Martineau, UMBC
Jim Mayfield, HLTCOE / APL
Scott Miller, BBN
Aaron Phillips, Carnegie-Mellon
Andrew Philpot, USC/ISI
Christine Piatko, HLTCOE / APL
Lane Schwartz, Univ. of Minnesota
David Zajic, Univ. of MD

Acknowledgment: This work is supported, in part, by the Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

Human Language Technology Center of Excellence
810 Wyman Park Drive
Baltimore, Maryland 21211
410-516-4800,
www.hltcoe.org

Semantically Informed Machine Translation (SIMT)

Kathy Baker, Steven Bethard, Michael Bloodgood, Ralf Brown, Chris Callison-Burch,
Glen Coppersmith, Bonnie Dorr, Wes Filardo, Kendall Giles, Anni Irvine, Mike Kayser,
Lori Levin, Justin Martineau, Jim Mayfield, Scott Miller, Aaron Phillips, Andrew Philpot,
Christine Piatko, Lane Schwartz, David Zajic



human language technology
center of excellence

Final Report of the 2009 Summer Camp for Applied Language Exploration
Document Last Updated: November 2, 2009

Contents

Chapter 1

Introduction

This report describes the findings of the machine translation team from the first Summer Camp for Applied Language Exploration (SCALE) hosted at the Human Language Technology Center of Excellence located at Johns Hopkins University. This intensive, eight week workshop brought together 20 students, faculty and researchers to conduct research on the topic of Semantically Informed Machine Translation (SIMT). The type of semantics that were examined at the SIMT workshop were "High Information Value Elements," or HIVEs, which include named entities (such as people or organizations) and modalities (indications that a statement represents something that has taken place or is a belief or an intention). These HIVEs were examined in the context of machine translation between Urdu and English. The goal of the workshop was to identify and translate HIVEs from the foreign language, and to investigate whether incorporating this sort of structured semantic information into machine translation (MT) systems could produce better translations.

The SIMT SCALE differs from other efforts in MT, most notably the DARPA Global Autonomous Language Exploitation (GALE) initiative. The key differences are:

1. The SIMT SCALE focused on incorporating syntax and semantics into machine translation whereas linguistically naive approaches to MT have dominated much of the GALE research. Although syntactic translation models have not shown dramatic improvements in GALE's Arabic-English translation task, we found that they dramatically improve Urdu-English translation.

2. The SIMT SCALE worked on translation for a low-density language, which has minimal amount of bilingual training data. In GALE, hundreds of millions of words worth of bilingual texts are used to train statistical translation models. In SCALE, only 1.5 million words of Urdu-English texts were available.

These differences created novel research directions for the SIMT SCALE, and resulted in promising findings that suggest that syntax and semantics may improve machine translation quality for other low-resource languages. In addition to an examination of the usefulness of linguistic information for low resource machine translation, other aspects of the SIMT project that may be of value to the defense and research communities include: improved translation of HIVEs by identifying such elements and providing special handling for them; a reduction of MT errors (e.g., mistranslated or dropped entities and modalities) in the face of sparse training conditions; and a variety of strategies for overcoming data sparsity problems and for acquiring additional data.

1.1 Problems translating low resource languages

It is informative to look at an example translation to understand the challenges of translating important semantic entities when working with a low-resource language pair. Figure 1.1 shows an example taken

Figure 1.1: An example of Urdu-English translation. Shown are an Urdu source document, a reference translation produced by a professional human translator, and machine translation output from a state-of-the-art system before the SIMT SCALE.

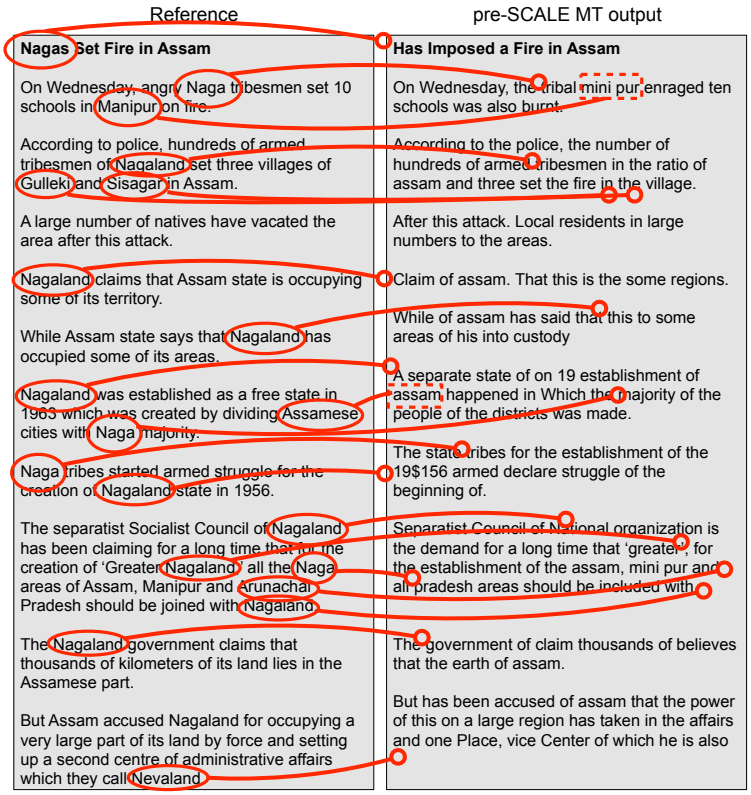


Figure 1.2: Translation Errors due to Missing or Incorrect Named Entities

from the 2008 NIST Urdu-English translation task, and illustrates the translation quality of a state-of-the-art Urdu-English system (prior to SIMT SCALE). The system was trained on 38,000 sentence pairs containing 900,000 Urdu words and 880,000 English words. The small amount training data for this language pair results in signicantly degraded translation quality compared, e.g., to an Arabic-English system that has more than 100 times the amount of training data. The machine translation output in Figure 1.1 was produced using Moses, a state-of-the-art phrase-based machine translation system that does not incorporate any linguistic information like syntax or morphology or transliteration knowledge. As a result, it is unable to translate words that were not directly observed in the bilingual training data. Names, in particular, are problematic. For example, the lack of translation for *Nagaland* induces multiple omissions throughout the translated text. Other names like *Manipur* are transliterated poorly (as *mini pur* in the first sentence above). Figure 1.2 highlights the full set of translation errors due to missing or incorrect named entities.

1.2 High Information Value Elements (HIVEs)

For machine translation to be successful it must convey the meaning of the high information value elements contained in the foreign source document. HIVEs are by their nature important for the comprehension of the foreign documents. Translating HIVEs accurately is important whether the machine translation system is used by a human analyst or by a downstream application like an information extraction program or knowledge base. The two types of HIVEs that were the focus of the SCALE workshop were *named entities* and *modalities*. We investigated whether the integration of these HIVEs into MT would enable higher precision in translation of units of information conveying *who* and *where* as well as the certainty of *events*, i.e., whether they really happened.

The types of named entities of interest to us included PERSON (*John*), ORGANIZATION (*New York Times*), LOCATION (*New York*), and TITLE (*Mr.*). Identifying entities could allow specially handling in order to eliminate prevalent errors that occur in statistical translation with sparse training data. As seen in Figure 1.2, most named entities get dropped. Moreover, most commonly used automatic evaluation measures for MT fail to assign a heavy penalty to a missing or incorrect person name. It is important to find these and translate them correctly so that they do not go undetected.

The types of modality of interest to us include, among others: *firm belief true* (e.g., John is in New York); *firm belief not true* (e.g., John is not in New York); *requirement true* (e.g., John should be in New York); *belief may be true* (e.g., John might be in New York); and *wants true* (e.g., John wants to be in New York). The idea behind the identification of modality is that these could constrain the translation output in ways that would not otherwise be captured correctly, e.g., we might wrongly predict that John is in New York (with certainty) if *might* is not adequately conveyed during the translation. Moreover, most standardized MT measures fail to assign a heavy penalty to a missing negation marker (e.g., *not*). More details about each of these HIVE types are provided in Chapter 8.

Better translation of named entities and modalities would allow us to understand statements about who did what to whom, and to understand whether such information is stated as beliefs, intents, or facts.

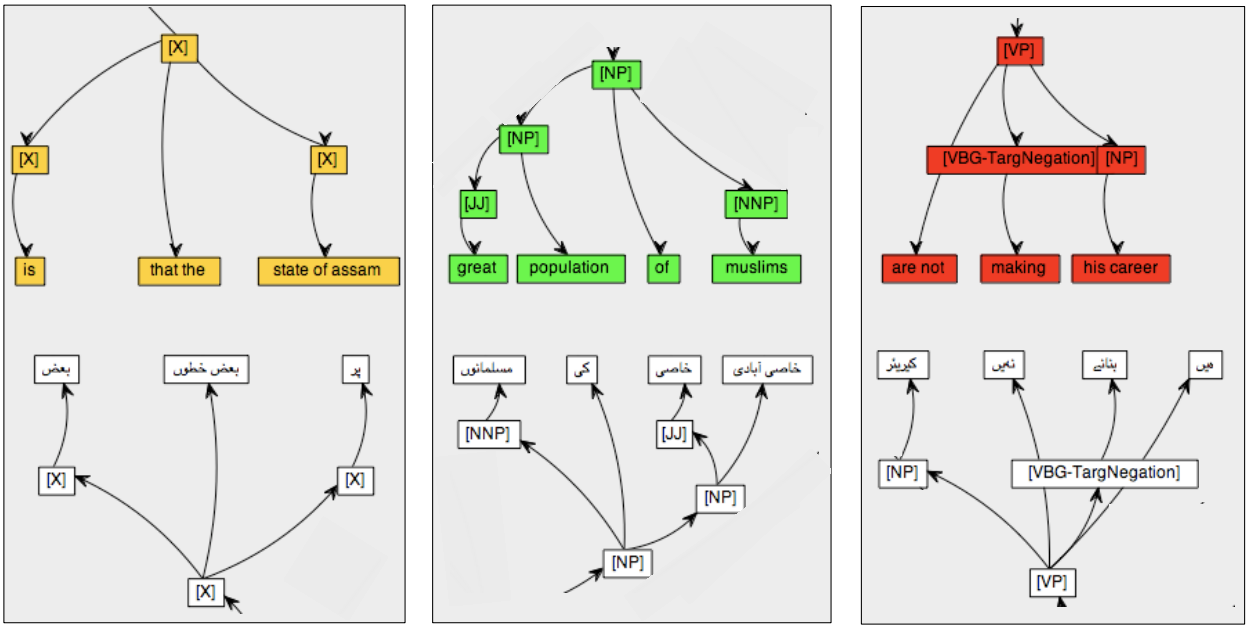
1.3 Integrating HIVES for Semantically Informed Machine Translation

The mission of the SIMT SCALE was to incorporate semantic information such as HIVEs into machine translation. A critical component of this mission was to create new ways of integrating structured information into machine translation systems. We created general structured models of machine translation that were not only capable of incorporating semantic elements, but also made effective use of syntactic information. In our workshop, there were two primary ways of incorporating structured information like HIVEs into machine translation:

- HIVEs were used as higher-order symbols inside the translation rules used by the translation models. Generic symbols in translation rules (like the generic non-terminal symbol "X") were replaced with structured information at multiple levels of abstraction. Examples of the structured information folded into translation rules include grammatical categories, named entities, and modalities
- HIVEs were used to augment the input sentence prior to application of MT. By identifying HIVEs in input sentences, they could be sent to specialized processing modules rather than translated in the normal fashion. For example, names were handled by a specialized transliteration module.

We incorporated HIVEs into two different machine translation systems which employed different approaches to translation. The first machine translation system was a hierarchical phrase-based model which originally used only a single, generic non-terminal symbol "X", which was subsequently replaced with more structured information. Figure 1.3 illustrated the evolution this machine translation system (the Joshua decoder) by replacing "X" with grammatical categories and then with HIVE categories. The second machine translation was an example-based MT system. This system (Cunei) was extended to use the HIVE annotations and lexical clustering to search the training corpus to generate phrasal translations that match parts of the input sentence. Lexical clustering provided a means to expand the phrase query and locate syntactically or semantically similar translations of the input. HIVE annotations helped to determine how to piece the partial phrasal translations together into a larger structure.

We demonstrated that structured models can dramatically improve the quality of translation for low-density languages like Urdu. Over the summer the quality of our machine translation system improved by 6 Bleu points on a blind test set administered by NIST, matching the highest score reported on the Urdu-English task. See Chapter 2 for details. Chapter 3 shows improvements to Cunei.



Standard Hierarchical Rules | Syntactic Enhancements | Semantically Informed Rules

Figure 1.3: The evolution of HIVE integration in the Joshua decoder. At the start of summer the decoder used translation rules with a single generic non-terminal symbol, later syntactic categories were used, and by the end of the summer the translation rules included semantic entities such as modalities.

1.4 Other Strategies for Improving Translation of Low Resource Languages

Beyond structured modeling for machine translation, we explored a variety of other strategies to cope with the data sparsity problems that are faced when working with a low-resource language pair like Urdu-English. This report describes a number of such strategies:

- Chapter 4 discusses a method for discovering parallel sentences from texts that are comparable, but that are not translations of each other. The method works by indexing newswire documents in two languages (in this case documents from the English BBC and from the Urdu-language version of the BBC World Service) using an information retrieval system. A baseline MT system is used to translate Urdu sentences into English. These sentences are used as queries to the information retrieval system, and relevant English documents are retrieved. A support vector machine classifies sentences from the retrieved documents as being likely translations or not. Likely translations are paired with their original Urdu sentences, thus increasing the amount of bilingual parallel data available to train statistical translation models.

- Chapter 5 investigates methods for inducing a bilingual dictionary from monolingual texts. Monolingual corpora are used to build vector-space semantic representations of English words and of foreign words based on their co-occurrence with other words in those languages. A seed bilingual dictionary is used to project across the vector space representations for the two languages. A vector for an unknown foreign word is projected into English space. Candidate translations are discovered by comparing the cosine distance of the projected vector against vectors representing all English words. We show how this method can be generalized to more than two languages, and how dimensionality reduction can be used to create a shared space representing concepts across all languages by projecting from a high dimsional space onto a low dimensional one.

- Chapter 6 explores efficient ways of using people who are fluent in both English and Urdu. Rather than have them create bilingual training data by translating randomly selected sentences, we use active learning techniques to select sentences that will be most useful to the model when translated. Selective sampling makes better use of an annotator's time and therefore improves the quality of the model faster than random sampling. We explore methods to increase the vocabulary and decrease the number of unseen bigrams, trigrams and 4-grams. We further explore having people translate short phrases instead of just sentences. Whereas previous work performed simulated active learning by pretending that the bilingual corpus is not yet translated, we conducted non-simulated experiments by hiring Urdu-English translators through Amazon's Mechanical Turk online labor market.

- Chapter 7 shows how to train a transliteration model using the statistical machine translation pipeline. Instead of training data consisting of pairs of translated sentences, we train the transliteration model on Urdu names paired with their transliteration into English. The model learns a mapping between sequences of Urdu letters with their English equivalents. We detail a two-pass model that does a rough transliteration into English and then automatically post-edits the output. The transliteration model is integrated into the machine translation system to deal with out of vocabulary words that were previously untranslatable. We acquire training data for the transliteration model in three ways: by tagging English names in the parallel corpus and extracting their corresponding Urdu segments, by crawling English Wikipedia articles about people and gathering those that link to an equivalent Urdu article, and by gathering large numbers of transliterated names using Mechanical Turk.

1.5 Results of the SIMT SCALE

The most significant result of the SIMT SCALE was the integration of linguistic knowledge (both syntax and semantics) into statistical machine translation in a unified and coherent framework. By augmenting hierarchical phrase-based translation rules with syntactic labels that were automatically extracted from a parsed parallel corpus, and further augmenting the parse trees with semantic elements such as named entities and modality markers, we produced a better model for translating Urdu and English. The resulting system significantly outperformed the linguistically naive baseline model, and reached the highest scores yet reported on the NIST 2009 Urdu-English translation task. This finding has particular importance because it shows linguistically informed machine translation helps for a language pair where only a relatively small amount of bilingual training data is available, and the source and target languages have significantly different word order. The Urdu-English challenge had both characteristics. In our experiments, the ability of syntactically-informed translation rules to generalize word reordering resulted in the largest Bleu score gains (over 3 Bleu points for integrating syntax). We expect that a similar result will hold for other low resource languages that have significantly different structure than English.

Other significant aspects of the SIMT SCALE were the following:

1. A preliminary exploration of modality was performed which produced a detailed taxonomy of modality types. Further work should explore how accurately modality can be tagged monolingually, and how to integrate modality taggers with knowledge bases.

2. Solid work was performed in name transliteration which produced excellent results. Careful measurements were made to determine the impact of various conditions.

3. New active learning methods were explored, showing how performance could be ramped up quickly with little annotation effort. These experiments were non-simulated, in contrast to most active learning research where experiments are simulated by holding back the labels on the existing training corpus,

and revealing the labels as the algorithm selects the items. Several interesting variations were tried, careful measurements were made, and clear quantitative gains were observed.

4. To overcome the sparse lexical coverage of the small bitext, several interesting attempts were made to induce latent structure from monolingual and comparable corpora. Results show promise but are not yet conclusive.

Looking forward, there appear to be significant opportunities to further refine the syntactic and semantic entities used in our translation framework. This could include a exploring wider range of semantic relations or automatically subcategorized syntactic types. Moreover the characteristics of Urdu-English that made syntactic machine translation so effective are likely to be achieved other language pairs. The SIMT SCALE has equipped its participants with skills to further develop the science, technology, and tools for efficient machine translation in a low-density environment. In particular, they will continue to innovate the architectures, models, and metrics for more structured and robust machine translation models.

Chapter 2

Joshua System Overview

Joshua (Li et al., 2009) is an open source decoder[1] for statistical machine translation that uses synchronous context free grammars (SCFGs) as its underlying formalism. SCFGs provide a convenient and theoretically grounded way of incorporating linguistic information into statistical models of translation. Linguistic information is especially useful when only limited bilingual training data is available, and when the source language is significantly different in word-order from the target language, as is the case with Urdu-English. In this section, we describe improvements that were made during the SCALE summer workshop to Urdu-English machine translation using syntactically- and semantically-informed models. We got significant gains in translation quality – 6 Bleu points on a blind test set constructed by NIST – over a state-of-the-art baseline system that does not incorporate linguistic information.

2.1 Motivation

Phrase-based (Och and Ney, 2002; Koehn et al., 2003) and hierarchical phrase-based approaches (Chiang, 2005; Chiang, 2007) to statistical machine translation have dominated the field for the past half decade. While such approaches to statistical machine translation (SMT) have shown considerable success in DARPA's GALE program, there are questions about whether they are appropriate when such large volumes of training data are not available. The quality of phrase-based and hierarchical phrase-based SMT relies on memorizing phrases that are observed in the training data. Phrase-based models have little capacity for generalization, and are unable to learn even simple linguistic facts. These approaches result in poor estimates of translation probabilities under small data conditions and for morphologically rich languages, since they are plagued by sparse counts. Importantly, phrase-based and hierarchical phrase-based models contain no explicit linguistic information and therefore cannot learn even simple generalizations like that a language's word order is subject-object-verb or that adjective-noun alternation occurs between languages.

During the SCALE summer workshop, we chose to investigate incorporating syntactic and semantic information into our Urdu-English translation. We believed that this strategy would likely be successful since the Urdu-English language pair has only a very small amount of training data compared with the languages that are the focus of the GALE program (its parallel corpus is less than 1% the size of the GALE training data), and because Urdu's word order is radically different than English. These factors combined mean that the rote memorization strategy, as taken by phrase-based translation, is unlikely to be successful.

Over the past few years, a wide variety of strategies have been used to try to incorporate syntax into statistical machine translation (Melamed, 2004; Quirk et al., 2005; Galley et al., 2006; Cowan et al., 2006; Marcu et al., 2006; Liu et al., 2006; Zollmann and Venugopal, 2006; DeNeefe et al., 2007; Cherry, 2008;

[1]The decoder can be downloaded at `http://www.sourceforge.net/projects/joshua`, and the instructions in using the toolkit are at `http://cs.jhu.edu/~ccb/joshua`.

	Urdu	English
S →	NP① VP②	NP① VP②
VP→	PP① VP②	VP② PP①
VP→	V① AUX②	AUX② V①
PP →	NP① P②	P② NP①
NP →	*hamd ansary*	*Hamid Ansari*
NP →	*na}b sdr*	*Vice President*
V →	*namzd*	*nominated*
P →	*kylye*	*for*
AUX →	*taa*	*was*

Figure 2.1: A toy example that illustrates a SCFG that can translate (romanized) Urdu into English for one sentence.

Mi et al., 2008; Zhang et al., 2008; DeNeefe and Knight, 2009). No consensus has been reached about what the best approach is for incorporating syntax, nor has it been definitely shown that syntactic information improves over more conventional phrase-based approaches to statistical machine translation.

We selected the synchronous context free grammar (SCFG) formalism for a number of reasons:

- It is a theoretically grounded formalism that allows syntax to be incorporated into translation.
- Extracting SCFGs from a parallel corpus only requires parses on one language's side, so we did not need to develop an Urdu parser.
- We were able to extend the Joshua decoder so that it was able to handle arbitrary SCFGs, rather than being limited to Hiero-style grammars, which have a single non terminal symbol "X".

2.2 Synchronous Context Free Grammars

SCFGs generalize context free grammars so they generate pairs of related strings. Because they generate pairs of strings they are useful for specifying the relationship between two languages, and can be used to describe translation and re-ordering. Probabilistic SCFGs can be formally defined as follows:

- T_S: a set of source-language terminal symbols
- T_T: a set of target-language terminal symbols
- N: a shared set of nonterminal symbols
- A set of rules of the form $\mathrm{X} \rightarrow \langle \gamma, \alpha, \sim, w \rangle$
  - $\mathrm{X} \in \mathrm{N}$
  - γ is a sequence source terminals and non-terminals
  - α is a sequence of target terminals and non-terminals

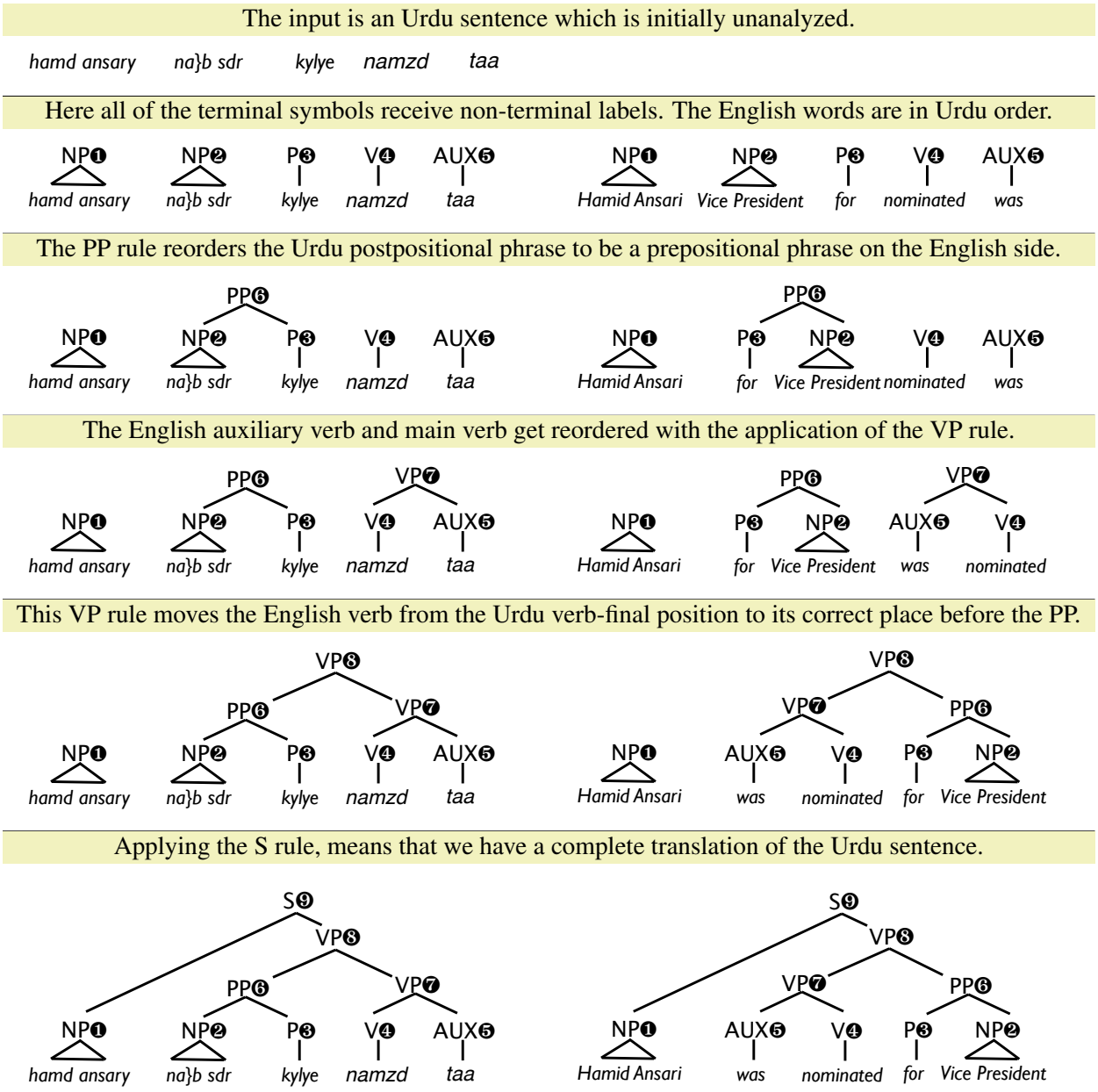


Figure 2.2: Using SCFGs as the underlying formalism means that the process of translation is one of parsing. This shows how an English sentence can be generated by parsing the Urdu sentence using the rules given in Figure 2.1

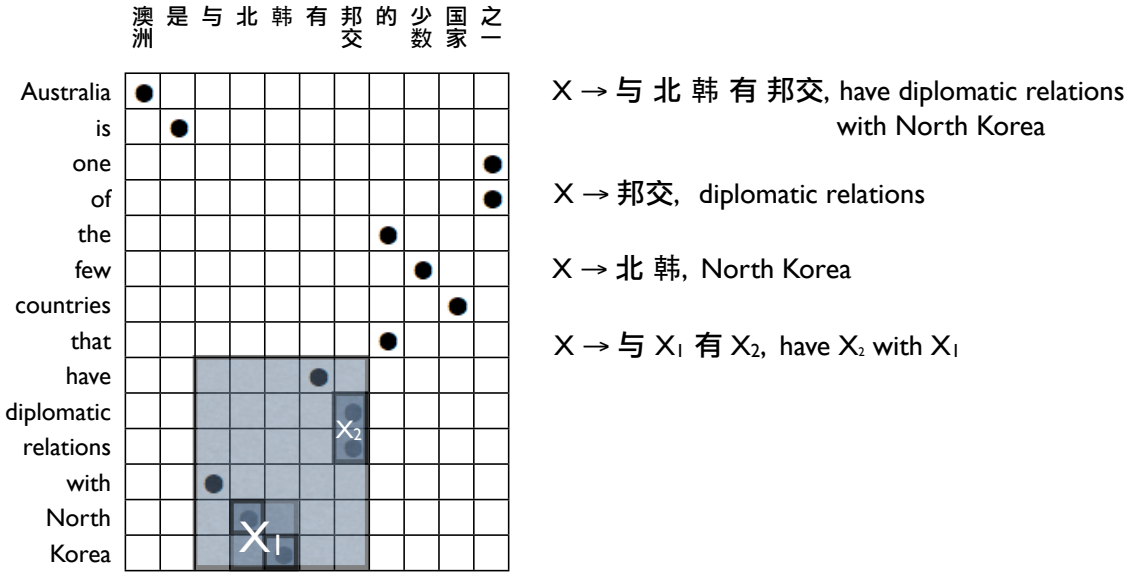


Figure 2.3: An example of a hierarchal phrase extracted from a word-aligned Chinese-English sentence pair. Chiang's Hiero system used rules that were written in the form of synchronous grammars, but which are devoid of linguistic information.

- $\sim$ is a one-to-one correspondence between the non-terminals in γ and α
- w is a weight or probability assigned to the rule

A toy example of an SCFG is given in Figure 2.1. The nonterminal symbols, which are written in uppercase, are identical across the two right hand sides of the context free grammar rules, but can come in different orders. The process of translation is accomplished by parsing the source language input sentence and simultaneously generating the target language output sentence. This process is illustrated in Figure 2.2, which shows how parsing an Urdu sentence generates an English translation using the toy example grammar. The toy grammar and example parse omit the w weights/probabilities assigned to the grammar rules. In practice there are a huge number of alternative translations and derivations, and assigning probabilities allows us to choose the best translation according to the model, and to reduce the search space by expanding only the most promising partial translations.

2.2.1 Hiero: SCFGs without linguistic non-terminals

The use of SCFGs for statistical machine translation was popularized by Chiang (2005) with the introduction of the Hiero system. Chiang's Hiero system extended the standard phrase-based approaches to statistical machine translation by allowing phrases that contain gaps. Chiang described how these *hierarchical phrases* could be obtained by straightforwardly extending the standard methods (Koehn, 2004; Koehn et al., 2003; Tillmann, 2003; Venugopal et al., 2003) for extracting phrase pairs from word-aligned sentence pairs. Figure 2.3 shows how a hierarchical phrase can be constructed by replacing two smaller phrase pairs with the nonterminals X_1 and X_2. These rules are written in the same format as the SCFG rules given in Figure 2.1. Although the Hiero rules are devoid of linguistic information, they are able to indicate reordering, as shown with the swapped positions of X_1 and X_2.

Rather than using the full power of the SCFG formalism, the Hiero system instead uses a simple grammar with one non-terminal symbol, X, to extend conventional phrase-based models to allow phrases with

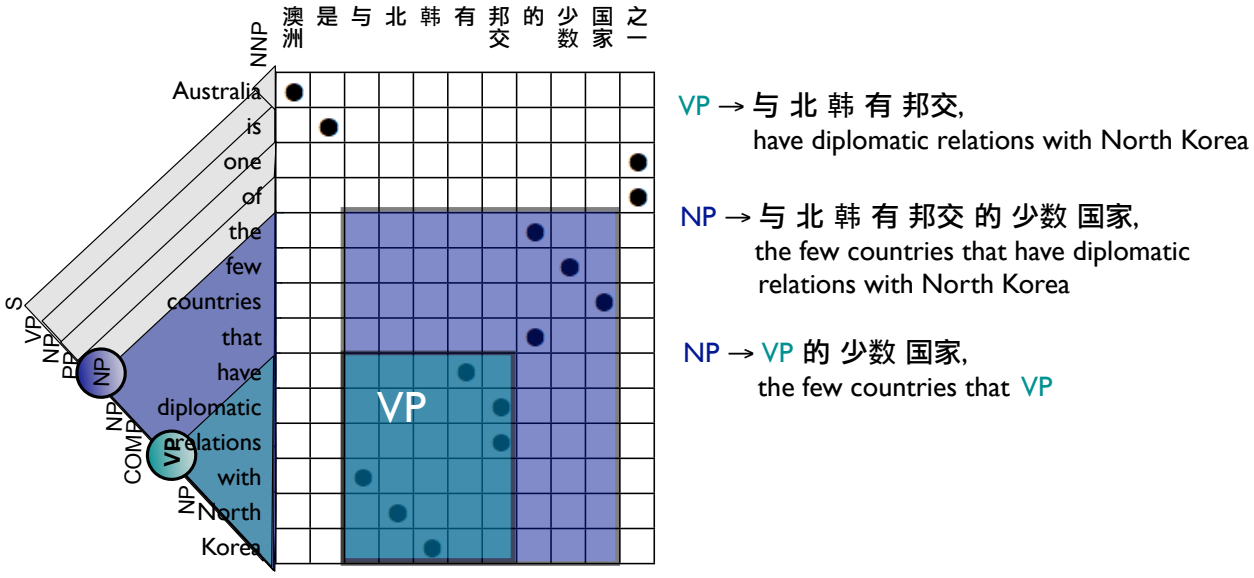


Figure 2.4: An example of extracting linguistically-motivated SCFGs by labeling phrase pairs using the labels of the corresponding nodes in a parse tree. Rules with syntactic non-terminals on the right-hand sides can be formed by replacing phrase pairs with their non-terminal labels, as shown with the VP on the right hand sides of the second NP rule.

gaps in them. The Hiero system is technically a grammar-based approach to translation, but does not incorporate any linguistic information in its grammars. Its process of decoding is also one of parsing, and it employs the Cocke-Kasami-Younger (CKY) dynamic programming algorithm to find the best derivation using its probabilistic grammar rules. However, because Hiero-style parses are devoid of linguistic information, they fail to capture facts about Urdu like that it is post-positional or verb final.

2.2.2 SCFGs with syntactic non-terminals

The Joshua decoder was originally a re-implementation of the Hiero system described in Chiang (2007). In preparation for the SCALE summer workshop we modified the Joshua decoder to accept any SCFG instead of just Hiero-style grammars. We were able to leverage a tremendous amount of the decoder's existing machinery including its CKY chart-parsing algorithms, its n-gram language model integration, its beam- and cube-pruning, its k-best extraction algorithms and its minimum error rate training (Och, 2003) module.

Joshua now can process synchronous context-free grammars with a rich set of linguistically motivated non-terminal symbols. In order to use such grammars, we must first have a way of extracting them from a bilingual parallel corpus. A number of approaches have been proposed for extracting linguistically-motivated grammars from a parsed parallel corpus (Galley et al., 2004; Zollmann and Venugopal, 2006). Figure 2.4 shows how linguistically motivated grammar rules can be extracted from a training sentence pair that has been automatically parsed and word-aligned. Instead of assigning the generic label "X" to all extracted phrase pairs, as the Hiero model does, we use linguistic labels from the parse tree. Note that one of the major advantages of extracting the linguistic SCFG for an automatically parsed parallel corpus is that only one side of the parallel corpus needs to be parsed. To extract an Urdu-English SCFG we therefore could use an English parser without the need for an Urdu parser. During translation the Urdu input text gets parsed with the projected rules, but a stand-alone Urdu parser is never required.

In the standard phrase-based and hierarchical phrase-based approaches to machine translation, many of

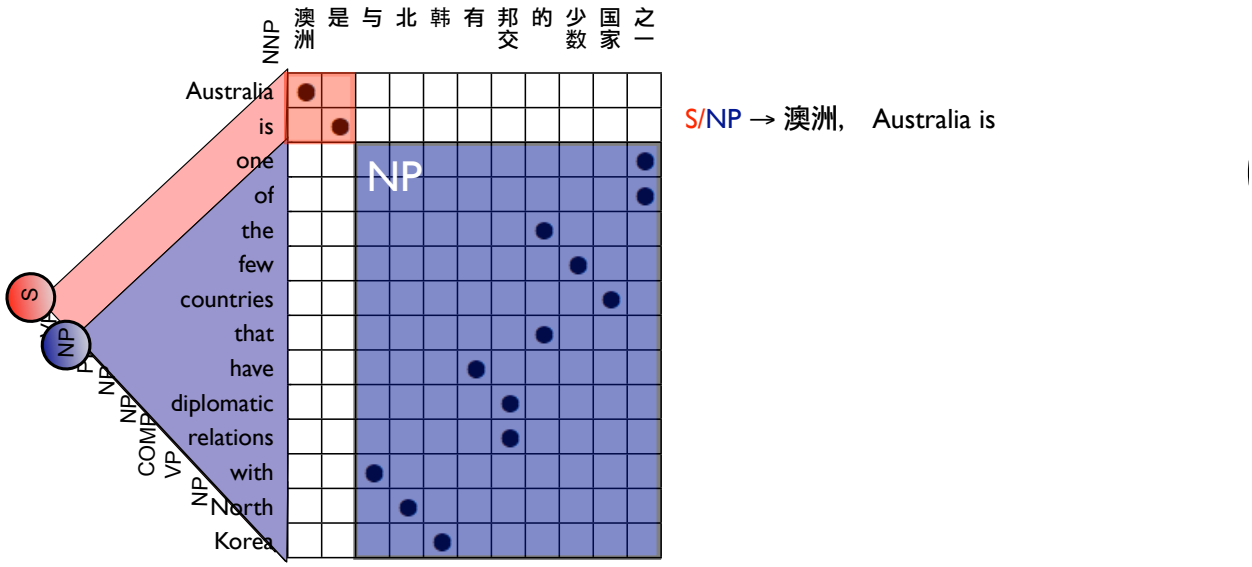


Figure 2.5: An example of a CCG-style label for the non-constituent phrase *Australia is*. The label indicates that the non-constituent phrase would be an S if an NP were found to its right. This complex label can be treated like any other label during parsing and translation.

the "phrases" that are used are not phrases in the sense of syntactic constituents. They are simply sequences of words. For example, in Figure 2.4, the phrases *Australia is* and *one of* are examples of phrases which have consistent Chinese phrase pairs, but which do not correspond to nodes in the parse tree. If we were to limit ourselves to extracting only phrase pairs that were licensed by the parse tree and which had consistent word alignments, then we would have reduced coverage compared to Hiero. Instead, we adopt the methodology described byZollmann and Venugopal (2006), which achieves the same coverage as Hiero by generating complex labels for non-consistuent phrases.

Zollmann and Venugopal (2006) define a framework for extracting SCFGs with linguistically motivated nonterminals from an aligned parallel corpus where one side of the parallel corpus has been parsed. The resulting context-free rules contain a rich set of nonterminal categories. These categories include traditional nonterminal categories taken directly from the monolingual parse trees (e.g. DT, NP, VP), and extended categories formed by gluing together adjacent nonterminals (e.g. NP+V, RB+JJ) and incomplete constituents that denote a category missing a child component to the left (e.g. NP\DT) or to the right (e.g. NP/NN) in the style of Combinatory Categorial Grammars (Ades and Steedman, 1982). Zollmann and Venugopal (2006)'s Syntax Augmented Machine Translation (SAMT) grammars may also include hierarchical rules and glue rules (Chiang, 2007) that are not linguistically motivated; such rules allow partial translations to be combined (with some penalty) without regard to syntactic structure.

For the experiments that we conducted over the summer, we used the grammar extraction software that comes as part of their open source SAMT toolkit[2] (Venugopal and Zollmann, 2009). We modified the Joshua software to accept grammars in the format output by the SAMT package. Joshua translates by applying the extracted SCFG rules to the source language text using a general chart parsing framework. This general approach enables Joshua to use SCFGs that contain only hierarchical phrase-based rules (Chiang, 2007) as well as SAMT grammars that incorporate linguistic information (Zollmann and Venugopal, 2006). Chart parsing results in a hypergraph (Huang and Chiang, 2005) that represents a shared forest of source

[2] `http://www.cs.cmu.edu/~zollmann/samt/`

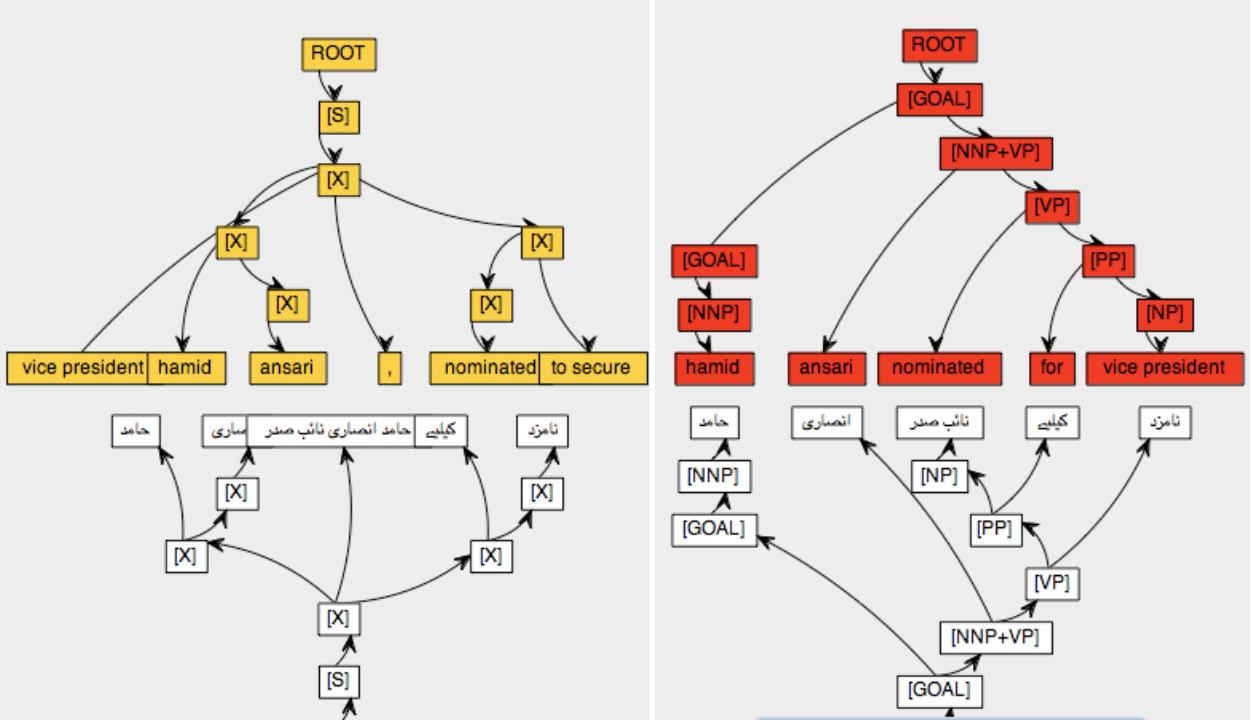


Figure 2.6: Sample derivations for an Urdu sentence translated into English translation using a hierarchical phrase-based SCFG (left) and a linguistically motivated SCFG (right). For reference, a human translator produced "Hamid Ansari nominated for the post of vice president" as the translation of the Urdu.

side parses, and the corresponding target side translations. Figure 2.6 shows an example of the one-best derivation of an Urdu sentence translated into English, using a hierarchical phrase-based SCFG and using an SAMT grammar.

2.3 HIVE-augmented Grammar Rules

In addition to extracting syntactic grammar rules, we also extracted grammar rules with *semantic* entities containing the high-information value elements (HIVEs) described in Section 1.2. The syntactically-informed grammar extraction procedure requires parse trees for one side of the parallel corpus. While it is assumed that these trees are labeled and bracketed in a syntactically motivated fashion, the framework places no specific requirement on the label inventory. We take advantage of this characteristic by providing the rule extraction algorithm with augmented parse trees containing syntactic labels that have HIVEs grafted onto them so that they additionally express semantic information.

Our strategy for producing augmented parse trees involves three steps:

1. The English sentences in the parallel training data are parsed with a syntactic parser. In our work, we used the lexicalized probabilistic CFG parser provided by Basis Technology Corporation.

2. The English sentences are tagged with HIVEs using one or more information-extraction systems. We used the Phoenix tagger to mark the sentences with named entities, and and a modality tagger developed at the workshop to mark modality information.

3. The semantic HIVEs are grafted onto the syntactic parse trees using a tree-augmentation procedure. The grafting procedure was implemented during the workshop and is inspired by Miller et al. (2000).

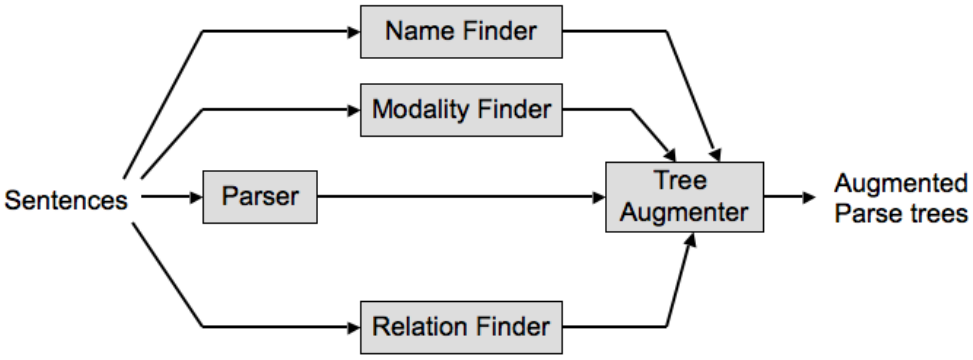


Figure 2.7: Workflow for producing HIVE-augmented parse trees. The English side of the parallel corpus is automatically parsed, and also tagged with HIVEs such as named entities, modality makers, and other relations. The HIVE tags are then grafted onto the syntactic parse trees.

The workflow for producing HIVE-augmented trees is illustrated in Figure 2.7.

Figures 2.8 and 2.9 show, respectively, a parse tree and the HIVE annotations for the sample sentence. The HIVEs in this example include named entities as well as descriptions and relationships. We note that while our framework is general, we incorporated only named entity and modality HIVEs during the 8-week workshop; descriptions and relationships are anticipated as future work.

A 5-step procedure suffices for grafting names, descriptions, and relationships onto syntactic parse trees:

1. Nodes are inserted into the parse tree to distinguish names and descriptors that are not bracketed in the parse. For example, the parser produces a single noun phrase with no internal structure for "Lt. Cmdr. David Edwin Lewis". Additional nodes must be inserted to distinguish the description, "Lt. Cmdr.," and the name, "David Edwin Lewis."

2. Semantic labels are attached to all nodes that correspond to names or descriptors. These labels reflect the entity type, such as person, organization, or location, as well as whether the node is a proper name or a descriptor.

3. For relations between entities, where one entity is not a syntactic modifier of the other, the lowermost parse node that spans both entities is identified. A semantic tag is then added to that node denoting the relationship. For example, in the sentence "Mary Fackler Schiavo is the inspector general of the U.S. Department of Transportation," a co-reference semantic label is added to the S node spanning the name, "Mary Fackler Schiavo," and the descriptor, "the inspector general of the U.S. Department of Transportation."

4. Nodes are inserted into the parse tree to distinguish the arguments to each relation. In cases where there is a relation between two entities, and one of the entities is a syntactic modifier of the other, the inserted node serves to indicate the relation as well as the argument. For example, in the phrase "Lt. Cmdr. David Edwin Lewis," a node is inserted to indicate that "Lt. Cmdr." is a descriptor for "David Edwin Lewis."

5. Whenever a relation involves an entity that is not a direct descendant of that relation in the parse tree, semantic pointer labels are attached to all of the intermediate nodes. These labels serve to form a continuous chain between the relation and its argument.

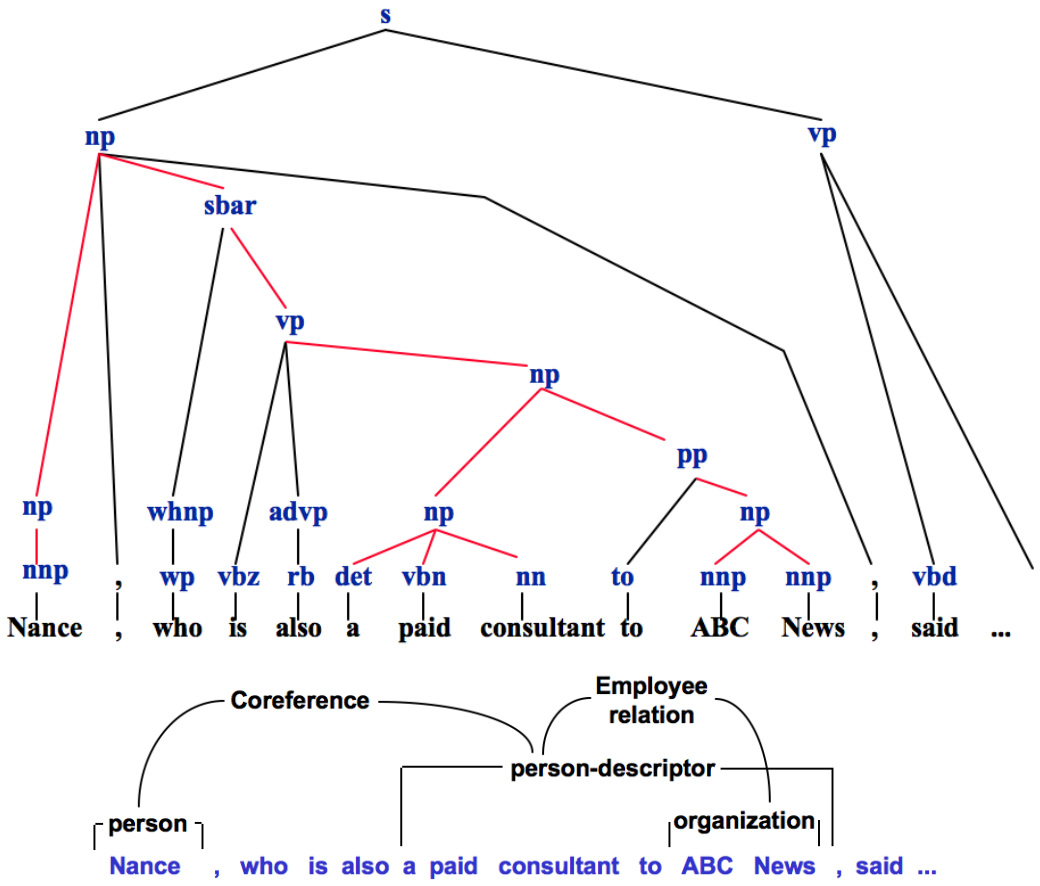


Figure 2.8: A sentence on the English side of the bilingual parallel training corpus is parsed with a syntactic parser, and tagged with a name finder.

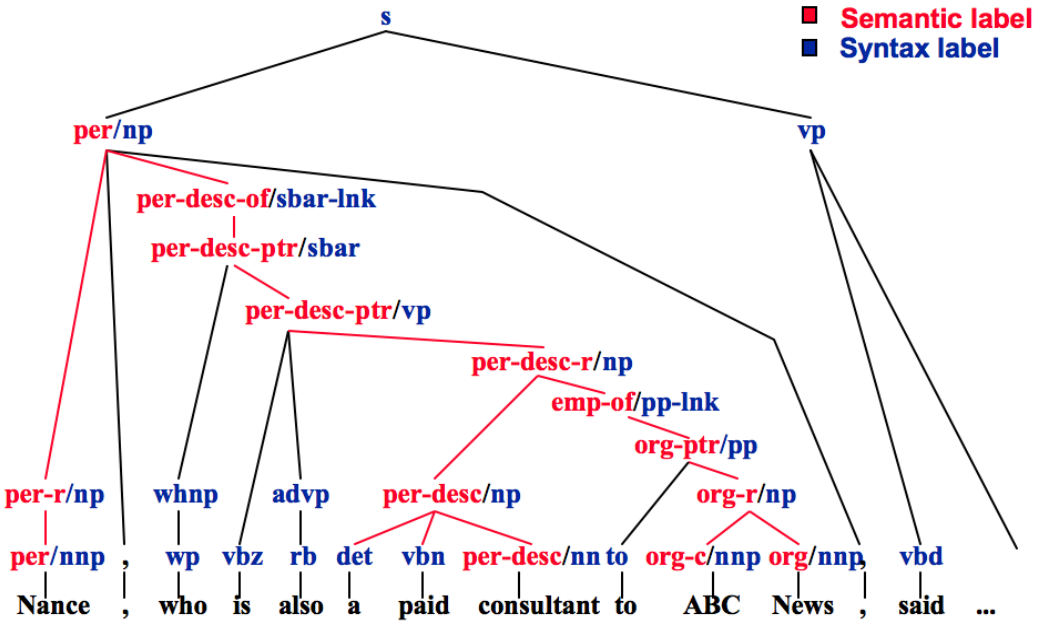


Figure 2.9: The name tags are grafted onto the syntactic parse tree prior to grammar extraction.

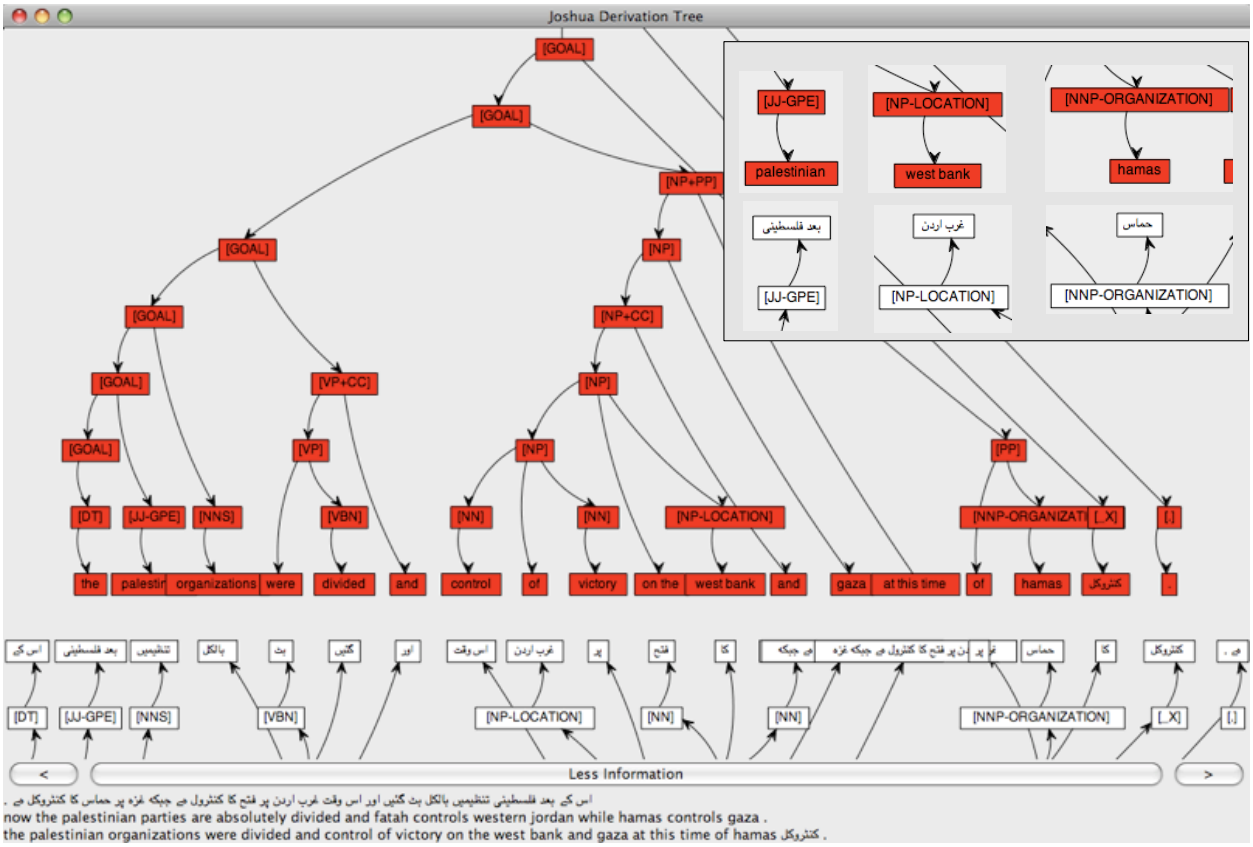


Figure 2.10: Example translation constructed from grammar rules that include named entities grafted onto syntactic nonterminals. The inset window highlights three of the entity tags. Because labels are simultaneously generated on the Urdu side, the translation system acts as a very simple Urdu named entity tagger.

Once augmented trees have been produced for the parallel corpus, the trees are presented, along with word alignments (produced by an aligner such as GIZA++), to the SAMT rule extraction software to extract synchronous grammar rules that are both syntactically and semantically informed. These grammar rules are used by the Joshua decoder to produce translations. Figure 2.10 shows an example translation that includes grammar rules with named entities, and Figure 2.11 shows an example translation that includes modalities. Because these HIVEs get marked on the Urdu source as well as the English translation, semantically enriched grammars that include HIVEs also act as very simple named entity and modality taggers for Urdu. However, only names and modalities that occurred in the parallel training corpus are marked in the output.

2.4 Experimental Setup

This section details the data and software that we used. The results are given in Section 2.5.

2.4.1 Data Sets

Training data The Urdu-English bilingual parallel corpus was provided by the Linguistics Data Consortium.[3] The parallel corpus was re-processed by Wade Shen of MIT Lincoln Labs. The corpus consisted

[3]LDC catalog number LDC2009E12, "NIST Open MT08 Urdu Resources"

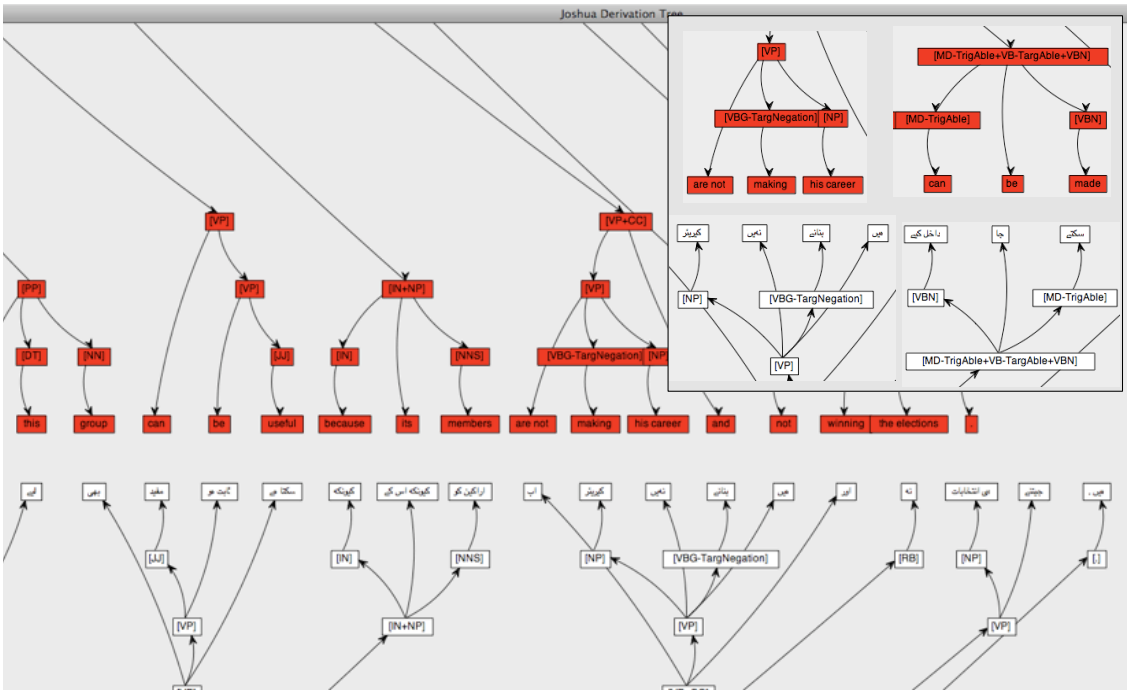


Figure 2.11: Example translation constructed from grammar rules with modalities grafted onto syntactic nonterminals. The inset window shows modalities tags from sentence and from another another sentence.

of 88,108 sentence pairs with 1,586,065 English tokens and 1,664,409 Urdu tokens. The LDC additionally provided a bilingual dictionary with 113,911 entries with 115,752 English tokens and 117,071 Urdu tokens. The number of unique words (types) in the combined dictionary and parallel corpus was 58,993 for English and 54,281 for Urdu. After normalization the number of types decreased to 50,720 for English and 53,459 for Urdu.

The monolingual English data used to train the language model was taken from the data from the DARPA GALE Arabic-English task. We re-used the large 5-gram English language model that was trained for the JHU GALE system. This was primarily based on the English Gigaword.[4] The trained language model consisted of 32,577 unigrams, 326,770 bigrams, 375,388 trigrams, 430,405 4-grams and 448,493 5-grams.

Tuning set We used a tuning set to set the models of our parameters using minimum error rate training. The tuning set consists of a set of Urdu sentences with multiple reference English translations for each Urdu source sentence. We split the 2008 NIST Urdu-English test set[5] into two pieces: one for tuning model parameters, and one for ongoing validation experiments. These two sets were referred to as "dev" and "devtest", respectively. Because the 2008 NIST test set contained two genres (newswire and web), we kept the balance between these when splitting into dev and devtest sets, so that both had the same proportion of newswire and web documents as the original 2008 NIST test set.

The dev set contained 981 sentence pairs. The tokenized source sentences contained 20,835 Urdu words with a vocabulary of 4,005 normalized Urdu types. The Urdu sentences were translated into English by four LDC-hired translators, however two of these translators produced nearly identical outputs so effectively there are only 3 reference translations per source sentence. The number of English words in the references

[4]LDC catalog number LDC2007T07, "English Gigaword Third Edition"
[5]LDC catalog number LDC2009E11, "NIST Open MT 2008 Current Test Set Urdu-to-English"

varied from 18,415–18,777. Their vocabulary sizes varied from 3,560–3,653 normalized English types.

Test sets During the workshop we evaluated our experiments on the second part of our split of the 2008 NIST Urdu-English test set, which we referred to as the devtest set. The devtest set contained 883 sentence pairs. The tokenized source sentences contained 21,623 Urdu words, with a vocabulary of 4,120 normalized Urdu types. The number of English words in the multiple reference translations varied from 19,150–19,864. Their vocabulary sizes varied from 3,636–3,782 normalized English types.

At the end of the summer, we evaluated our best performing system on the 2009 NIST Urdu-English test set, which had been kept blind during the workshop. We referred to this data as the test set. The test set contained 1,792 sentence pairs. The tokenized source sentences contained 42,358 Urdu words, with a vocabulary of 5,628 normalized Urdu types. The four English references translations contained between 37,842–41,319 English tokens and 4,915–5,131 normalized English types.

2.4.2 Software

Tokenization and normalization For English, we used a tokenizer that followed the Penn Treebank specification and normalized the text by lowercasing it. For Urdu, we used a tokenizer and a normalizer that were developed by Jim Mayfield during the SCALE workshop (see Appendix C).

Word alignment The word alignments in most of our experiments were produced using the Berkeley Aligner (DeNero and Klein, 2007)[6]. For a handful of experiments we tried combining the results of multiple word aligners by separately aligning the parallel corpus with each of them and then combining the word-aligned parallel corpora together, effectively multiplying out the length of the corpus. Experiments that refer to "multiple alignments" used the Giza++ (Och and Ney, 2003)[7], the Basis syntactic aligner and Chris Dyer's Hadoop aligner (Dyer et al., 2009) in addition to the Berkeley aligner.

Grammar extraction Grammar extraction for the Hiero-style grammars was performed with Joshua's built-in rule extraction module. This module is based on Adam Lopez's suffix array grammar extractor (Lopez, 2007), and was re-implemented by Lane Schwartz. Grammar extraction for the syntactically-informed SCFGs was performed using the grammar extraction module of the open source SAMT toolkit (Venugopal and Zollmann, 2009).[8]

Language modeling We held the language model constant across all of the experiments reported here. We trained a 5-gram language model on the LM data available to the GALE program. In particular the data included the English GigaWord, but left out the UN data. The model was trained using the SRI Language Modeling Toolkit (Stolcke, 2002).[9]

Decoder The decoder used for the experiments reported in this section was version 1.2 of the Joshua decoder (Li et al., 2009).[10]

[6] `http://code.google.com/p/berkeleyaligner/`
[7] `http://code.google.com/p/giza-pp/`
[8] `http://www.cs.cmu.edu/~zollmann/samt/`
[9] `http://www.speech.sri.com/projects/srilm/`
[10] `http://cs.jhu.edu/~ccb/joshua`

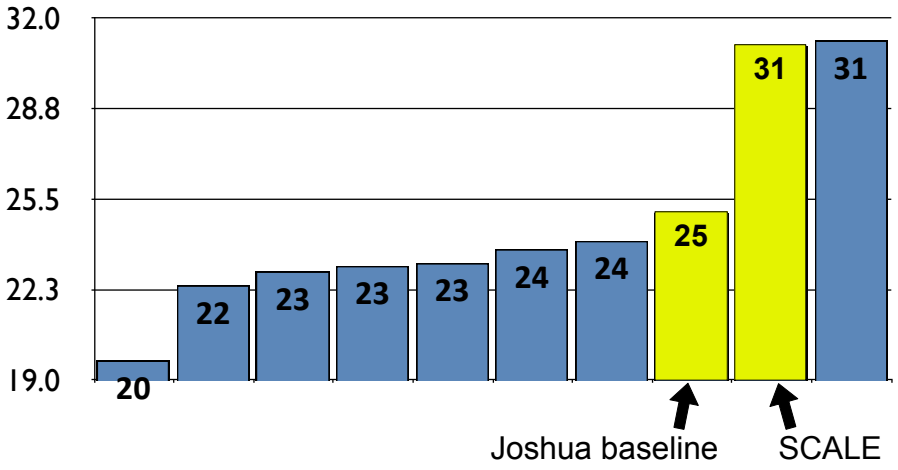


Figure 2.12: Performance of all submissions to NIST 2009 Urdu-English task (scores are cased Bleu). The arrows highlight the scores for the pre-SCALE baseline Joshua system that used the Hiero model and the final SCALE system using the syntax and semantics augmented system.

Recasing For experiments on the devtest set, we report results leaving the English output lowercased and tokenized. For experiments on the test set, NIST required re-cased, de-tokenized output. We re-cased using the Joshua decoder by training a "translation model" from lowercased English to cased English, and using a 5-gram cased English LM.

Scoring We evaluated our translation output with several automatic evaluation metrics: Bleu, Meteor, TERp, and HATERp (Papineni et al., 2002; Banerjee and Lavie, 2005; Snover et al., 2009). To simplify the interpretation of the results, we report only Bleu scores here. We used two scoring scripts: the Joshua implementation of Bleu scoring for experiments on the devtest set, and the NIST scoring script for the test set.[11]

2.5 Experimental Results

Figure 2.12 shows the performance of the pre-SCALE baseline system and the final SCALE system, along with all of the other systems entered into to NIST 2009 Urdu-English task (following the NIST guidelines, we anonymize systems other than our own). The figure shows that the final SCALE Joshua system made radical improvements in Bleu score over the baseline Joshua system from the start of the summer. The final system increased by 6 Bleu points over the already strong baseline. This reflects the fairest analysis of the final system because the NIST data set was kept blind to the SCALE participants and was only used to evaluate the system improvements at the very end of the summer. As a result of these improvements, the final system is as good as the best Urdu-English system.

The best performing system in the NIST 2009 Urdu-English task also employed a syntactic translation model, but is a closed source proprietary system. This re-inforces our finding that syntactic models are crucial for doing well when translating from languages with significantly different word order than English, which do not have copious amounts of bilingual parallel training data.

[11] `ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl`

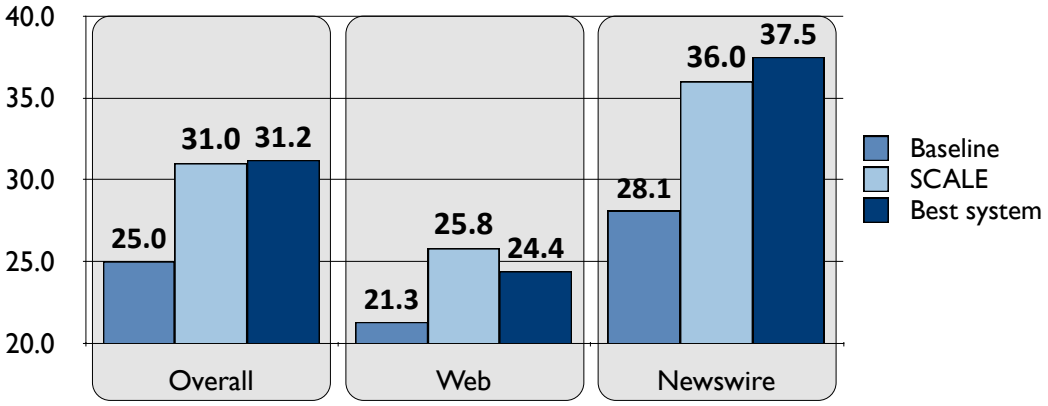


Figure 2.13: A breakdown of the performance of three systems on the Newswire and Web portions of the NIST 2009 test set along with Overall score.

Figure 2.13 gives a more detailed breakdown of the Bleu scores on the two genres in the NIST 2009 test set. The 0.2 Bleu points difference in the overall score between the best performing system and the SCALE system is not a statistically significant difference. When broken out into different genres, the closed-source syntactic system does 1.5 Bleu points better on the Newswire genre, and the SCALE system does better 1.4 on the harder Web genre. No conclusions should be drawn from these differences, since no adaptation was done during the SCALE workshop to improve on either of these genres in particular. If the parameters of the SCALE system were tuned for each genre specifically, then it might have achieved a higher score than the other system.

2.5.1 Example output

Figure 2.14 shows example translations of an Urdu article from the NIST 2009 test set. It shows translations produced using the baseline Hiero model that the Joshua decoder used prior to the SCALE workshop, and using the final SCALE system that used a SCFG enriched with syntax and semantics. A human translation is also given for reference. The headline of the article demonstrates the importance of syntax. The baseline system fails to put the verb in the correct English position:

> 'first nuclear experiment in 1990 was'

The SCALE system produces output in the proper English order:

> The First Nuclear Test Was in 1990.

In some sentences, the improved word order clarifies who did what to whom. For example in the last sentence of the baseline system's output, it is unclear who is providing the nuclear technology to whom:

> He said that China, North Korea, Iran, Syria, Pakistan, through Egypt, Libya and Yemen is to provide nuclear technology.

In the SCALE system it is clear that China is providing the technology to the other countries:

> He said that China would provide nuclear technology to North Korea, Iran, Syria, Pakistan, Egypt, Libya and Yemen.

In some case the lack of syntax guidance results in the baseline system outputting relations that are exactly wrong. For instance, the baseline system states that:

> He said as China and India was joint enemy of Pakistan.

When it should say that China and Pakistan have a common enemy in India. The SCALE final system does not express this exactly correctly, but it gets much closer:

> He said in response to China, Pakistan and India as a common enemy.

The final SCALE system also notably has far fewer untranslated Urdu words in its output. This improvement is due to its transliteration component. Two of the automatic transliterations are correct and make the system's output more understandable: Los *Alamos* National Laboratory, and *Nevada* desert. Others are close, but not quiet right: *Diana Steelman* instead of *Danny Stillman*. Several of them are incorrect in a way that are misleading: enrichment plant in *Cairo* instead of in *Karaj*. These are examined in more detail in Chapter 7.

2.5.2 Details of experiments

Figure 2.15 gives the results for a number of experiments conducted during the summer.[12] The experiments are broken into four groups: baselines, syntax, semantics, and other improvements. To contextualize our results we experimented with a number of different baselines that were composed from two different approaches to statistical machine translation – phrase-based and hierarchical phrase-based SMT – along with different combinations of language model sizes and word aligners. Our best performing baseline was a Hiero model with a 5-gram language model and word alignments produced using the Berkeley aligner. The Bleu score for this baseline on the development set was 23.1 Bleu points.

Our initial foray into incorporating syntax into the translation models resulted in a significant drop in Bleu score. The reason for this was that we directly re-used the translation model feature functions employed under our Hiero model. Those three feature functions (the probability of the English phrase given the Urdu phrase, the product of lexical translation probabilities from English to Urdu, and lexical probabilities from Urdu to English) were not sufficient to distinguish between good and bad translation rules. As we added additional feature functions into the linear model, and retrained it using minimum error rate training, the syntactic model improved. The feature 12 functions that we used in the best performing of the syntactic systems were:

- a **glue rule** feature that applies a penalty when the model backs off from the syntactic rules to the Hiero-style "X" rule
- a **rule application counter** which helps to control how many rules are applied
- a **target word counter** which allows a weight to be set so that it rewards or penalizes rules that have more target words.
- **-log(frequency/undiluted result count)**, a negative log (base e) result-conditioned relative frequency score
- **-log(simple frequency/undiluted result count)**, another result- conditioned relative frequency score with a simpler estimator of the frequency

[12] These experiments were conducted on the devtest set. The Bleu score for these experiments is measured on uncased output, which in general should be higher, but the devtest effectively had only three reference translations, which explains why the scores are lower than the scores on the NIST 2009 test set.

Baseline	SCALE	Reference
'first nuclear experiment in 1990 was' Thomas red Unilever National Laboratory of the United States in ویپن designer, are already working on the book of Los ایموس National Laboratory ینڈی, former director of the technical انٹیلی چنس written with the cooperation of سٹیمن. This book 'nuclear express: political history and the expansion of bomb' has been written, and the two writers have also claimed that the country has made nuclear bomb is he or any other country's nuclear secrets to ے ای چ or that of any other nuclear power cooperation is achieved. Thomas Reid said in a news یسوای interview that in 1990 in the era of Benazir Bhutto China had the experience of Pakistan's first nuclear bomb. Thomas red said that on the basis of many reasons he was sure that China had the experience of Pakistan's first nuclear bomb. reasons in the bomb design and the China scientists mentioned During the conversation with Information. He further said that this was the reason that only two weeks in Pakistan in 1998 and within three days in response to India's nuclear experience to nuclear experiment was able to. Thomas red reminded that in 61 in Russia has suddenly nuclear experience and was in response to the United States were to experience began 17 days in despite the fact that the United States had the bomb from a long period. He further said that the nuclear bomb in 1998 that Pakistan may experience of what was he was made from very carefully and confidence was to meet on the Pakistani scientists. Thomas was red when this question that China has provided the nuclear technology to Pakistan, In response, He said as China and India was joint enemy of Pakistan. He said that it is also present proof of that Dr. Abdul Qadeer Khan after the Chinese design. apart from this, The ے یکی obtained documents in Libya were is also confirmed it from them. To another question whether the joint nuclear tests is common, He said in وےنونس in the US Open in the desert of اذایوان servants for Britain's nuclear experiment. He said that we are guesses also believed that Israel should also provide access to the results of the this experience. Thomas Reid and this question was also is China's program of nuclear proliferation. On this Thomas Reid said that in 1991 to set up in China انفہاس city of Iran nuclear installation instructions for the raw material, and ین ای زذ is provided. Besides this, China has provided of enrichment plant in Iran خرج also. He said that China, North Korea, Iran, Syria, Pakistan, through Egypt, Libya and Yemen is to provide nuclear technology.	**The First Nuclear Test Was in 1990.** Thomas red of the United States, the National Laboratory in designer are already working on the book of Los Alamos National Laboratory, former director of the technical intelligence, with the cooperation of Diana steelman wrote. This book under the title of the spread of nuclear expressway: the political history of the bomb and this has been written and the two writers have claimed that the country also has made nuclear bomb or any other country, Korea nuclear secrets, or any of the other nuclear power cooperation. Thomas Reid said in an interview to the news that Benazir Bhutto in 1990 in the era of China The first nuclear bomb test. Thomas Reid said that on the basis of several reasons to believe that China has the first nuclear bomb test. reasons of this bomb design and China information obtained during the conversation with the scientists. He further said that this was the reason that Pakistan only two weeks in 1998 and was able to nuclear experiment in response to India's nuclear experience within three days. Thomas Reid had reminded that Russia has sudden nuclear experiment in 61 and in response to the United States to experiment in 17 days. despite the fact that the United States had quite a long period from the bomb. He further said that the experience of the atom bomb Pakistan in May 1998, he was to make very carefully and confidence on Pakistani scientists. Thomas red when was this question that China has provided to Pakistan nuclear technology, he said in response to China, Pakistan and India as a common enemy. He said that there is evidence that Dr. Abdul Qadeer Khan has also used the Chinese design after did the documents from Libya in were also confirmed to them. On a question whether the joint nuclear tests is common, he said in Hanoi in the United States In the Nevada desert open detainees for Britain's nuclear experiment. He said that it also that Israel should also provide access to the results of this experience. Thomas red with this question was whether China's program of nuclear proliferation. Thomas red said that in the Iranian city of China with 491 in Spain, the raw material for nuclear installation instructions and provides. In addition to this, China has also provided the enrichment plant in Cairo. He said that China would provide nuclear technology to North Korea, Iran, Syria, Pakistan, Egypt, Libya and Yemen.	**"First Nuclear Experiment conducted in 1990"** Thomas Reed, who has worked as a weapon designer in Lawrence Livermore National Laboratory of America, has written this book in collaboration with Danny Stillman,the former Director of Technical Intelligence of Los Alamos National Laboratory. This book has been written with the title 'Nuclear Express:A Political History of the Bomb And its Proliferation,' and in this both the authors have claimed that any country that has made an atomic bomb has either stolen the nuclear secrets of another country or has had cooperation with some other nuclear power. Thomas Reed said in an interview to US News that in 1990, in the era of Benazir Bhutto, China had conducted the experiment of Pakistan's first nuclear bomb. Thomas Reed said that he is convinced on the basis of several reasons that China has conducted the experiment of Pakistan's first nuclear bomb. Those reasons include the design of the bomb and information obtained while talking to the scientists of China. He further said that this was the reason why in 1998, Pakistan was able to conduct a nuclear experiment just in two weeks and three days in response to India's nuclear experiment. Thomas Reed also reminded that in 1961 Russia suddenly carried out a nuclear experiment and it took 17 days for America to do the experiment in response to this, although America already had this bomb for awhile. He further said that the atom bomb, whose experiment was done in 1998 by Pakistan, was developed with extreme care and Pakistani scientists had full confidence in it. When Thomas Reed was asked if China had provided the nuclear technology to Pakistan, he replied that India was a common enemy of China and Pakistan. He said that the proof to this also exists in that Dr. Abdul Qadeer Khan used the Chinese design, and, apart from this, the documents retrieved from Libya afterwards also proved this. To another question as to whether it is usual to carry out nuclear experiments with others, he said that in 1990 America openly conducted a nuclear experiment for Britain in the desert of Nevada. He said that we may also presume that Israel, too, was given access to the results of this experiment. Thomas Reed was also asked whether China's nuclear proliferation program is active. On this, Thomas Reid said that since 1991, China has been providing raw material, instructions, and designs for the nuclear structure situated in Ispahan, a city in Iran. Besides this, China has also provided an enrichment plant to Iran in Karaj. He said that China has been providing nuclear technology to Iran, Syria, Pakistan, Egypt, Libya, and Yemen through North Korea.

Figure 2.14: Example translations from the pre-SCALE baseline system and the final SCALE system, with a reference human translation.

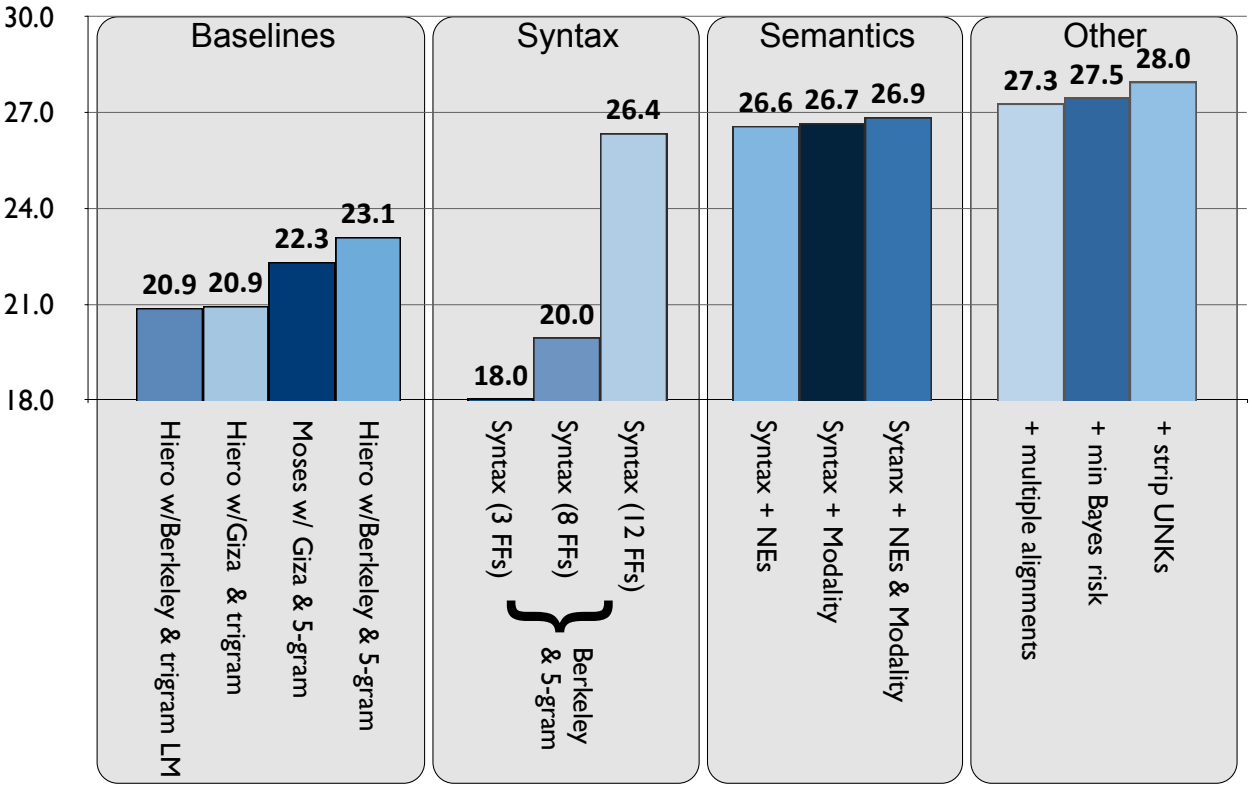


Figure 2.15: Results for a range of experiments conducted using Joshua during the SCALE workshop. The scores are lowercased Bleu calculated on the held-out devtest set.

- **-log(frequency/source and arg frequency)**, a negative log source-conditioned relative frequency score that does not take non-termial symbols into account
- a **non-monotonicity penalty** that governs how much re-orderings are penalizes.
- a **rareness penalty** which is meant as a bias against rare rules whose probabilities are unreliable because of sparse counts
- a collection of **four variants of the IBM Model 1** lexical translation probability score

These feature scores and others are automatically generated using the SAMT rule extraction software, and we selected a subset of the features. We found that adding more feature functions improved the Bleu score. A syntax model with 12 translation model feature functions outperformed the baseline Hiero model by 3.3 Bleu points – the largest improvements of any of our experiments.

After experimenting with syntactically motivated grammar rules, we conducted three experiments on the effects of incorporating semantic entities, aka HIVES, into the translation grammars. These were included by grafting named entities (NEs) and modality markers onto the parse trees, as described in Section 2.3. Individually each of these made modest improvements over the syntactically-informed system alone. Grafting named entities onto the parse trees improved the Bleu score by 0.2 points. Modalities improved it by 0.3 points. Doing both simultaneously had an additive effect and resulted in a 0.5 Bleu score improvement over syntax alone. This improvement was the largest improvement that we got from anything other than the move from linguistically naive models to syntactically informed models.

We performed three additional experiments that resulted in improvements. We moved from using a single set of word alignments produced by the Berkeley aligner to four sets of alignments from the Berkeley aligner, Giza++, the Basis aligner, and Chris Dyer's Hadoop aligner. We incorporated these in a simple fashoin by quadrupling the length of the parallel corpus, where each quarter had a different set of word alignments. This improved the Bleu score to 27.3 on the devtest set. We netted a small additional gain by using minimum Bayes risk re-ranking of the decoder's n-best output (Kumar and Byrne, 2004). Finally we got an additional 0.5 Bleu point gain by deleting any untranslated Urdu words. This final improvement was subsequently superseded by the workshop's transliteration effort.

2.6 Implications of Linguistically Informed MT

The take away message from these results is that significant improvements to translation quality be had through more linguistically informed structured models. By altering our synchronous context free grammars to have syntactically and semantically-informed rules, we made huge improvements to the quality of our Urdu-English translation system. Over the course of the eight week SCALE summer workshop, our system's Bleu score improved by a full 6 Bleu points, and was tied for the best-performance on the blind NIST 2009 Urdu-English test set. These improvements made clear the advantages of linguistically informed models for language pairs like Urdu-English, where only small amounts of training data is available and where the languages' structures are divergent.

Previous work into syntactically-informed machine translation has found mixed results, and has not found syntax to be as clearly useful as we found it to be. This is likely due to the fact that the advantage of syntax is lessened as huge volumes of training data become available – as with Arabic-English and Chinese-English where simpler memorization strategies are effective – or between languages which do not require significant amounts of reordering – as with French-English or Spanish-English. The implications of our findings are that syntactic information has the potential to radically improve the translation quality for low-resource languages with word order different from English. Thus these improvements are likely

to be transferable to languages like Korean and Farsi, as well as a host of others, which are low-resource languages with different word order.

Acknowledgements

We would like to acknowledge the efforts of four graduate students at JHU – Zhifei Li, Jonny Weese, Juri Ganitkevitch, and Wren Thornton – who were not on the SCALE workshop team but who worked intrepidly to provide support for the Joshua decoder before and during the workshop. We also extend our thanks to Wade Shen of MIT Lincoln Labs for re-aligning the parallel corpus, to Tim Anderson of AFRL for help with normalizing Urdu and suggesting using multiple aligners, and to Kay Peterson of NIST for keeping the test set blind and scoring our systems at the end of the summer.

Chapter 3

Cunei System Overview

Cunei (Phillips, 2007) is fundamentally a platform for data-driven machine translation. The system is 'trained' by indexing a (preferably large) corpus of bilingual text. Then at run-time the corpus is queried to locate examples of translations that are similar to the input. These translations may contain gaps or only cover part of the input sentence. Each translation is scored by a large number of feature functions. A statistical decoder searches through this space of possible, partial translations and combines them together in order to form a complete hypothesis that faithfully represents the input.

The advantage of a data-driven approach is that it is language-neutral and does not require a highly-skilled human to hand-code translation rules or lexical items. However, this also creates the problem that the system must induce the process of translation simply from a parallel corpus. Traditionally, the parallel corpora used during training only contains the source text and target text. In this work we extended the notion of a parallel corpus to allow for additional meta-information that can be used by Cunei at run-time. Initially we focused on meta-information that described semantic elements that we believed to be of high value to the translation process. Then this work was extended to seek broad coverage by generating annotations based upon automatic clustering. In both of these efforts the desire was to make more information about the parallel corpus available at run-time so that Cunei has a greater ability to correctly select and combine together the partial translations retrieved from the parallel corpus.

Cunei is open-source software and may be freely downloaded from `http://www.cunei.org/`.

3.1 Partial Structured Model

Of crucial importance to this work is *how* the additional meta-information is modeled and used during the translation process. As described previously, Joshua defines the translation process as a parse-derivation. The additional annotations were used to inform the parse tree and create more specialized, finer-grained nodes. In contrast, Cunei does not build one unified model for translation. Instead, as illustrated in Figure 3.1, Cunei models multiple sources of possibly divergent information. The 'partial structured model' built for each translation allows for each source of information to be modeled independently. In the illustration, if the blue circles are the lexical words, then the green, purple, and orange circles in the illustration represent alternative sources of information about the sequence. Perhaps the green circles are part-of-speech tags, the purple are dependency-structures, and the orange are person names. However, the circles have been left empty because the structure is intentionally flexible allowing for any type of meta-information. What defines a 'source of information' and its granularity is left to the user and may correspond to a specific type of annotation (such as 'person' or 'organization' identifiers) or what generated the annotation (such as Identifinder or Phoenix).

Each independent source of information is used to generate new feature functions that determine the

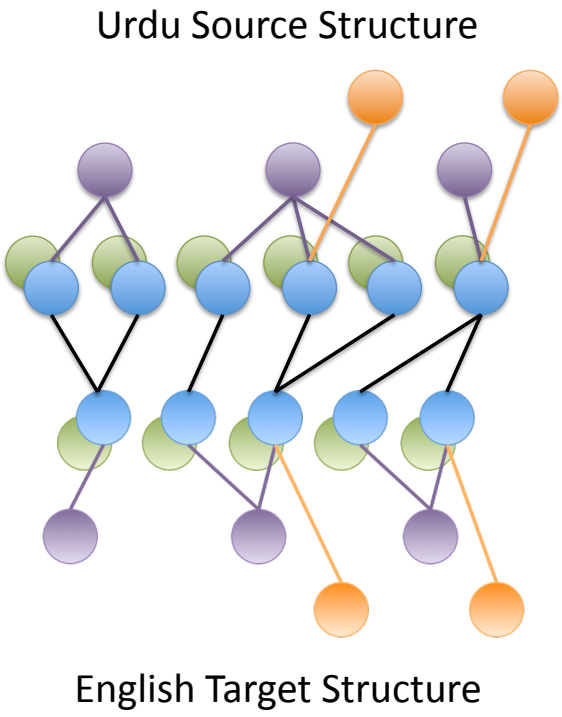


Figure 3.1: A Partial Structured Model for Translation

adequacy of a particular translation. On the source-side, the partial structure is used to determine how similar the retrieved translation is to the input that is desired to be translated. On the target side the partial structure is used as translations are combined together to ensure the coherency of the target hypothesis.

When Cunei retrieves translations from the corpus they may share similar structure (based on the additional meta-information) to the input, but not contain the same source words. Portions of the translation that do not contain the same source words as the input must be removed. The source words and corresponding target words (based on the alignment) are excised forming a template for translation which contains a gap. During decoding, hypotheses that correspond to the source words of the gap are used to complete the translation template. The partial structure of both the template and the substitution are used to favor substitutions whose structure is consistent with the template.

Lastly, Cunei maintains the partial structure of the source and target throughout the translation process. This provides the ability to produce richer output that contains the meta-information about the translation alongside the target text. We expect this particular capability of Cunei to assist downstream processes that consume the translations (either human or machine).

3.2 System Description

Cunei is distinguished from a traditional Statistical Machine Translation (SMT) system in that it delays as much computation as possible until run-time. In particular, translations are not retrieved from a pre-built phrase-table, but rather generated dynamically at run-time. This approach has three key advantages:

1. Run-time feature extraction makes it easy to model features dependent on the particular input or surrounding translations. For example, Cunei applies a feature to each translation indicating the number of words to the left and right that are not covered by the translation, but also matched the input. Similarly, contextual features can also be applied at the document level to measure the cosine or Jensen-Shannon distances between the input document and the document in the corpus in which the translation occurs.

2. Generating the translations at run-time produces a more consistent translation model. A traditional SMT system performs alignment and phrase-extraction once when the phrase-table is constructed. Typically this is done through a series of heuristics that determine whether two phrases may legitimately form a phrase-pair. Moses provides multiple heuristics for this very task with the default method known as 'grow-diag-final'. However, phrase-alignment is not in reality so neatly deterministic. For each source phrase in the parallel corpus, there is some probability (perhaps very small) that it translates to any given target phrase with which it has ever co-occurred. Cunei models the phrase extraction at run-time as part of the translation process with a series of alignment features that determine the probability that a given source phrase and target phrase are aligned. These alignment features are part of the final translation model and during each iteration of optimization the weights for these features are modified. The new weights change the probability distribution over the alignments and have the potential to extract different translations. Thus, the extraction and scoring of translations are not two processes, but form one consistent model.

3. This approach can efficiently search a larger space of possible translations. Phrase-tables quickly grow quite large–consuming gigabytes of disk space–but only a fraction of the phrases are ultimately used while translating a given test set. By generating translations on-the-fly, Cunei allocates all of its computation to phrases that are present in the input. Cunei first looks up translations for phrases that exactly, word-for-word match some span of the input. If the given phrase occurs frequently and Cunei believes it can model the translation well, then we move on. If this is not the case, then Cunei has the capability to extend the search space and retrieve similar phrases or generalized templates in order to better model the translation process.

The remainder of this section will describe in detail how translations are constructed at run-time.

3.2.1 Translation Selection

When given a new input to translate, Cunei searches the source-side of the corpus for phrases that match any sub-section of the input.

In order to accomplish this task, during training Cunei builds a suffix-array index for each type of sequence present in the parallel corpus. Minimally, the corpus will contain a lexical sequence representing the unaltered surface-strings (or less precisely, 'words'). Lemmas, part-of-speech, or statistical cluster labels may be used to generate alternative types of sequences. The suffix array for each type of sequence type is queried with the input to locate matches within the parallel corpus. A match may contain as few as one of the tokens from the input or exactly match the entire input. The collection of corpus matches is stored as a lattice with each element indexed by span of the input it covers.

The corpus matches are not required to be exact representations of the input. For example, a source phrase retrieved by matching only a part-of-speech sequence, may be structurally similar to the input, but it is likely to be semantically unrelated. Matches such as these do not 'as-is' provide valid translations, but they do still contain useful information about the translation process. For each token in a match that does not lexically match the input, a gap is formed. The gaps are projected to the target during phrase-alignment in order to form translation templates. Meta-information about the gap, such as the part-of-speech or HIVE

annotations, is preserved in order to aid selection of a valid replacement. These translation templates allow for the formation of novel phrases, but also add risk. As such, if exact phrasal translations are present, they are preferred.

For efficiency, not all matches present in the corpus are retrieved. Each match is scored individually by several feature functions to determine its relevance–how similar the match is to the input. Matches that have gaps are down-weighted. Matches that have the same context as the input (either sentential or document) are preferred. Typically, around a thousand matches are retrieved from the corpus for each span of the input, but only a few hundred are selected as the most promising and retained for alignment.

3.2.2 Translation Alignment

After a match is found on the source-side of the corpus, Cunei must determine the target phrase to which it aligns. The alignment is treated as a hidden variable and not specified during training. Instead, Cunei uses statistical methods to induce a phrase-alignment. Ideally, the full alignment process could be carried out dynamically at run-time. Unfortunately, even a simple word-alignment such as IBM Model-1 is too expensive. Instead, we run a word-aligner offline (GIZA++ and Berkeley are both supported) and use the word alignments as features for an on-line phrase-alignment.

Each source phrase has some probability of aligning to every possible target phrase within a given sentence. This probability is modeled in Cunei by a series of feature functions over the word-alignments.[1] When a source phrase is aligned to a target phrase, it implies that everything not in the source phrase is aligned to everything not in the target phrase. Separate features model the probability that the word-alignments for tokens within the phrase are concentrated within the phrase boundaries and that the word-alignments for tokens outside the phrase are concentrated outside the phrase boundaries. Additional features model whether to incorporate unknown words, lexicon-based translation probabilities, and the length ratio between the source and target phrase. This approach is modeled after the work of (Vogel, 2005) and (Kim et al., 2005).

For each instance of a match in the corpus, Cunei uses the feature functions to extract a scored n-best list of phrase-alignments. Each possible alignment forms an instance of translation between the source phrase and target phrase.

3.2.3 Translation Scoring

Of particular importance to Cunei's approach is that each *instance* of an Urdu-English translation from the corpus is scored individually. Most traditional SMT systems model a phrase-pair based on features that are maximum likelihood estimates over all instances of the phrase-pair in the corpus and treat each instance equally. Cunei, on the other hand, models each instance separately with its own log-linear model. This allows for features to be dependent on that particular instance of the translation in the parallel corpus. It explicitly models that some instances of translations are better than others. In the next step, all of the instances in the corpus that result in the same translation phrase-pair will be combined together with a single log-linear model for use during decoding.

Many of the per-instance translation features have already been discussed as they are usually calculated at the earliest stage in which the information is available. During matching, source-side similarity and contextual features are generated. These features are dependent both on the input and the particular sentence or document in the corpus in which the corpus match was found. During phrase-alignment, additional features

[1] Currently Cunei only uses one set of word-alignments which is assumed to be generated from the lexical sequence. One direction we considered extending Cunei during SCALE was to incorporate the additional syntactic and semantic knowledge directly into the on-line phrase-extraction. This should be a fairly straight-forward extension of the existing framework, but time did not permit us to explore it.

are calculated to measure the likelihood of the source phrase aligning to the target phrase *in this particular sentence*. Cunei also applies the traditional SMT features based on a translation's overall frequency in the corpus and a constant phrase penalty, none of which change from instance to instance. The scoring framework is intentionally flexible such that features are not hard coded and any number of features can be added to the translation model at run-time.

3.2.4 Translation combination

Thus far we have shown how each instance of a translation acquires several instance-specific features. However, we also need to take into account and explicitly model that some translations are only represented by a few instances in the corpus and some translations occur frequently. This is achieved by modeling collections of translation instances. The final log-linear model for a translation consists of a count feature indicating how many instances are present in the collection along with a series of constraints that indicate the minimum value for each instance-specific feature. A search is performed over each instance-specific feature such as the alignment quality, genre, or context to determine the set of constraint values that maximizes the score. Weights determined during optimization indicate the relative importance of each constraint and the match counts. Lower constraints will allow for more matches, while higher constraints will likely lead to higher quality phrase-pairs but with a smaller number of occurrences.

3.2.5 Optimization

Because at the end of the day, we use a log-linear model we can optimize it using the same Minimum Error Rate Training developed for SMT. In MERT, after each iteration, a new n-best list is generated with the optimized weights. Due to pruning and the beam search within the decoder, the new weights may yield different translations in the n-best list. In our approach, new weights can also change what translations are found in the corpus and how they are modeled. As a result, the search space is larger, but because we select the constraints that maximize the score, the model is still consistent and possible to optimize with MERT. Cunei's optimization code closely follows (Smith and Eisner, 2006) and is an improvement over traditional MERT and more appropriate for a large feature space.

3.3 System Improvements

Over the course of the summer we incorporated semantic knowledge into the translation process by adding semantic annotations to the parallel corpus and making these available to Cunei at run-time. The first phase concentrated on utilizing linguistically-motivated HIVE annotations. By definition, HIVE elements are believed to be of high-value and have a large impact on the semantic content of the resulting translation. As such, it is important to ensure that each HIVE is translated unambiguously. The additional HIVE annotations are designed to guide the translation process in selecting the most appropriate translation. In the second phase, we used the same framework we built that allowed for arbitrary HIVE annotations to be added to the corpus and extended this to any type of meta-information. In particular, we used automatic clustering to annotate the entire corpus. Several off-the-shelf clustering algorithms were used to generate these data-driven annotations that contained shallow syntactic and/or semantic knowledge.

3.4 Incorporating HIVE Annotations

The HIVE annotations used in this work came in two different flavors. Nouns were generally annotated if they represented an entity such as a person, organization, or location. Verbs, on the other hand, were

annotated if they expressed a specific modality. It would be unreasonable for a human to mark all of these annotations on the training corpus. Instead, we collected a small amount of human-annotated data and trained statistical classifiers to label the training corpus and input files.

In Cunei, each annotation appears as meta-information describing a specific span–either in the corpus or the input. These annotations are not required to form any particular structure; annotations may overlap and a given span may contain multiple annotations. Each of these annotations are associated with a specific annotation category as defined by the user. In this work we divided the annotations into two different categories: entities and modalities.

When an instance of a translation is found in the corpus, the annotations present in both the source and target-side of the translation are loaded as meta-information. If annotations on the source-side are available, they are compared to the input. For each annotation category, feature functions measure how many annotations are present in both the instance from the corpus and the input compared to how many annotations are only labeled on one of these. While the source-side annotations are used for retrieval and selection, annotations on the target-side are used when translations are combined together during decoding. When a translation predicts a gap being filled in by another translation, feature functions compare the annotations present on the gap to the annotations present on the translation filling the gap. Likewise, partial translations may predict that an annotation has begun, but is not yet complete. When two translations are concatenated, feature functions check how many annotations are now complete vs how many annotations remain incomplete. The feature functions on the target-side use the annotations to inform Cunei how to compose translations in a syntactic and semantically-coherent manner. The more weight assigned to these feature functions, the more Cunei will prefer to combine translations that have consistent annotations. Using a development set Cunei automatically tunes the weights for these annotation feature functions jointly with all other feature functions determining the best weights to use for translation.

It is preferred if meta-information is present on both the source-side and target-side of the parallel corpus. This gives Cunei the maximum flexibility. If a source of uninformative meta-information is used it will receive a weight of zero (or close to zero) during optimization and effectively be ignored. However, the system is not dependent on both source and target meta-information being present. In the absence of the source-side annotations, the target-side features will still be generated. Likewise, the opposite is also true and source-side annotations can be used without target-side annotations.

3.4.1 Examples

The process of using HIVE annotations during translation and generating HIVE-annotated output is shown in Figure 3.2 and Figure 3.3. Figure 3.2 illustrates two persons being identified in the input and Cunei forming a translation that correctly incorporates two person annotations. In particular, the entire phrase “benazir bhutto” is identified as one person so the translation system will prefer to generate translation that keep “benazir” and “bhutto” together as an entity. Figure 3.3 demonstrates the use of modality annotations. Our modality tagger was weaker in Urdu than English, and as such, the input sentence provided to Cunei was not properly marked with the ‘TrigAble’ and ‘TrigNegation’ modality annotations. However, during the translation process, Cunei saw enough evidence in the parallel corpus of partial translations using these modality annotations and it was able to properly recover these annotations on the English translation. The inverse is true with the two person annotations present on the Urdu in Figure 3.3. While Cunei properly translated the names, every time these two names occurred in the parallel corpus they lacked a person annotation. As such, Cunei followed the data and chose to believe that there should not be a person annotation present on the English translation. After seeing this result, we confirmed that the name finder we used consistently failed to identify these two English names in our parallel corpus. Thus, this particular issue is a problem in the name finder, but it is a humble reminder that data-driven methods are only as good as the data on which they are trained.

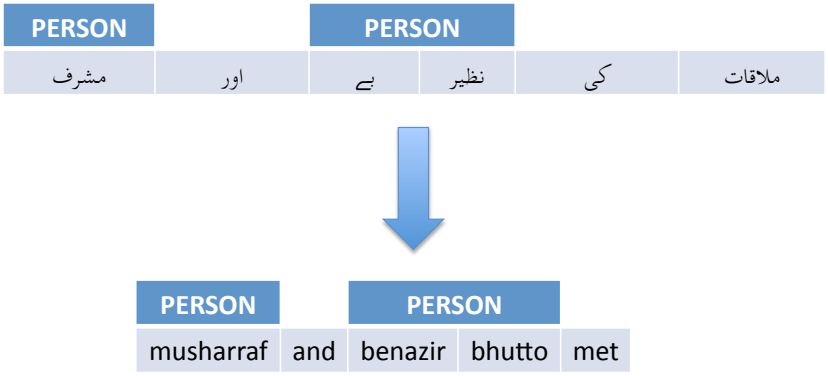


Figure 3.2: Example of Translation Using Annotations

3.4.2 XML Output

As mentioned previously, we believe that the additional semantic content carried within these annotations may additionally benefit downstream processes. Internally, Cunei preserved the annotations that it believes should exist on each translation. This information is readily available in XML as shown in Figure 3.4.

3.5 Learning 'Annotations' from Monolingual Data

Because it is much easier to obtain large monolingual corpora than large parallel corpora, we also investigated ways to leverage a large monolingual corpus in Cunei in addition to the parallel corpus used to generate actual translations. Three separate but related approaches to using monolingual corpora to permit generalization of the parallel corpus were investigated: syntactic clustering using the Brown algorithm, topic clusters using the LDA algorithm, and dynamic synonym clustering.

For Urdu, the BBC news archives provided approximately 20 times as much monolingual Urdu text compared to the bilingual corpus available from the LDC Urdu language pack. We used the text collected from the BBC as the monolingual source-language corpus for our experiments in augmenting the parallel corpus which are described below. Some experiments also used a portion of the even-larger English news archives from the BBC as a comparable corpus to the Urdu news stories.

3.5.1 Static Automatic Clustering

The meta-information framework we developed in Cunei was designed for integrating the HIVE annotations. However, we soon realized that the same framework could be used for any type of meta-information. To complement our work with HIVE annotations, we also used off-the-shelf clustering algorithms to apply automatically-induced annotations (cluster labels) to every token in the corpus. While likely not as informative as the human generated HIVE categories, these automatic clusters, induced from a large monolingual corpus, still add novel syntactic and/or semantic knowledge.

Unlike HIVE annotations, automatic clustering guarantees a cluster label for every token in the corpus. This unique property enables us to treat the cluster labels a different 'view' of the corpus and use the cluster labels to retrieve similar translations at run-time. The clustering algorithms are permitted to label each instance of a word differently. However, in our experiments with the Brown and LDA clusters, we enforced

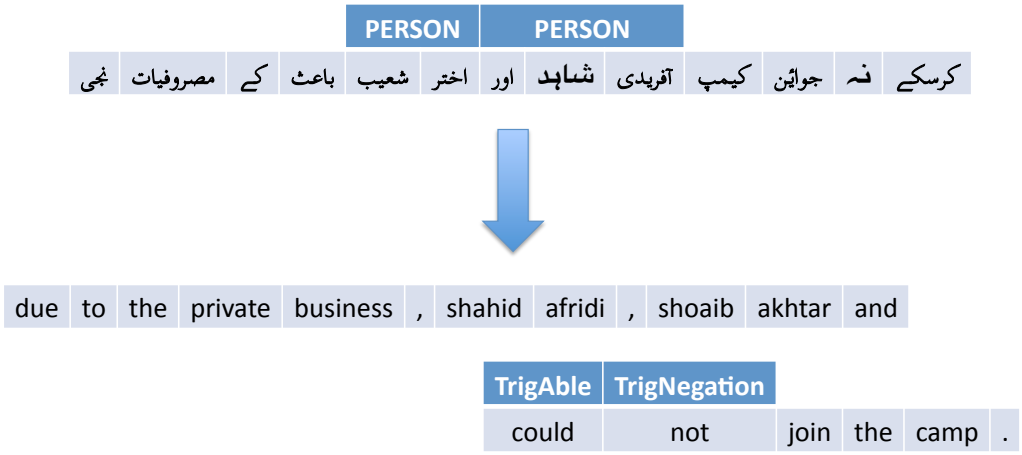


Figure 3.3: Example of Translation Using Annotations

that every token in the corpus is labeled with one and only one cluster label. This permitted Cunei to treat each mode of clustering as a distinct sequence of tokens. The sequences of cluster labels are indexed during training and then queried at run-time based on the clustered labels present in the input. The 'similar' translations are compared to the input and sections of the translation that do not lexically match are marked as gaps for substitution. However, these gaps contain annotations (minimally the sequence of cluster labels by which it was retrieved) that can be used to constrain the selection of translations for substitution.

Because the cluster label annotations are a special case of the general annotation framework used by the HIVE annotations, many of the feature functions for the cluster label annotations are the same. On the source-side, the clustered sequences for each instance of a translation are compared to the input and feature functions give preference to those that match. Likewise, on the target-side, when gaps occur, feature functions use the cluster labels in the same manner to prefer substitutions that are consistent.

One unique property of the cluster labels is that they never form 'incomplete annotations' because they only span one token. Thus, we do not generate feature functions on the target-side that prefer translations which form 'complete' cluster labels. Instead, we leverage the fact that every token has an annotation and build a language-model on the target-side sequence of cluster labels. For each mode of clustering we build a separate language-model. These language-model scores are used by feature functions to represent how to compose translations that are syntactically and semantically well-formed.

Lastly, for each mode of clustering we model a probability of translation from one cluster label on the source-side to one cluster label on the target-side. This is calculated exactly the same way as the lexical translation probabilities and uses the word-alignment links to count how many times each source cluster label aligns to each target cluster label. We are able to perform this calculation on the cluster labels because they are guaranteed not to overlap.[2]

114	suggest say insist accuse allege warn laud
162	whereby where encompassing wherein
188	floating spilling spewing playing rolling
303	believe think applaud belive detest reckon
329	theory verdict case indictment code lawsuit
423	korean african caucasian anatolian ossetian

Table 3.1: Example clusters derived from the Brown bigram mutual information clustering.

Brown Clusters

The first word clusters we experimented with were based on the class-based bigram language models of Brown and colleagues (Brown et al., 1992). In this paradigm, we build a bigram language model over words w_k and classes c_k such that:

$$Pr(w_k|w_{k-1}) = Pr(w_k|c_k)Pr(c_k|c_{k-1})$$

That is, the current word is conditioned on the current class, and the current class is conditioned on the preceding class.

Brown and colleagues showed that finding the best classes for this language model is equivalent to finding a mapping from words to classes such that the average mutual information of adjacent classes (e.g. c_k and c_{k-1}) is minimized. To produce such a clustering, the typical approach is to start with each word in its own cluster, and repeatedly merge the pair of clusters for which the loss in average mutual information is the lowest. The result of this process is a binary tree of clusters, which is typically trimmed at a certain depth to produce the desired number of word clusters.

We applied the Brown clustering algorithm to derive 1000 English clusters and 1000 Urdu clusters. We trained the language model on about 50 million words of each language, collected from the BBC. Some of the resulting clusters for English are shown in Table 3.1. These clusters are fairly typical of a bigram language model based clustering algorithm – words grouped together often have some semantic component in common (e.g. *suggest* and *say* are both verbs of speaking), but the clusters are strongly grammatical (e.g. all *-ing* verbs).

The class-based language models derived in this way were then used to assign cluster labels to each word in the Urdu and English training data, to each Urdu word in the test data, and to each word in the BBC English data. The cluster labels on the parallel training data allow the model to look up possible English translations of Urdu phrases using cluster labels instead of words, while the cluster labels on the BBC English data allowed us to build a language model[3] over the cluster labels to complement the usual language model over words.

Incorporating the Brown clusters demonstrated clear improvement over the baseline.

Figures 3.5 and 3.6 demonstrate Cunei using the cluster labels on actual sentences in our test set. In Figure 3.5, Cunei uses the cluster label sequence to find a translation in the corpus that is very similar to one fragment of the input. However, within this translation one of the lexical source words does not match, so a gap is formed on the source and the target removing 'government'. A translation for the missing word is found elsewhere in the parallel corpora and because it has the same cluster label, it forms a good substitution, composing a novel target phrase. This substitution worked well in part because the Brown clustering identified that 'government' and 'cabinet' frequently occur in the same syntactic structure. The

[2]We wanted to incorporate this calculation for all types of annotations (such as the HIVEs), but their lack of structure and potential overlap makes the problem much more complex and a good solution was not found.

[3]An SRILM 4-gram model with interpolation and Witten-Bell discounting

9	daily newspaper reported newspapers media
33	criticised strongly opposed saying supported
39	poor rich poverty class live wealth
87	trial court judge case evidence prosecution
307	party alliance national ruling democratic
584	fight terrorism war america terror enemy

Table 3.2: Example clusters derived from the LDA topic models.

same process takes place in Figure 3.6, except in this case the substitution is even more similar resulting in merely the change of verb tense.

LDA Clusters

To give a different cluster-level view of the data, we also experimented with cluster labels derived from Latent Dirichlet Allocation (LDA). Where the Brown bigram mutual information clusters consider only the preceding word and cluster label as context, LDA considers all the words in the sentence as context. This typically generates more semantic or topic-based clusters than the very syntactic clusters generated by the Brown clustering.

LDA (Blei et al., 2003) is usually presented as a generative model, as an imagined process that someone might go through when writing a text. This generative process looks something like:

1. Decide what kind of topics you want to write about.
2. Pick one of those topics.
3. Imagine words that might be used to discuss that topic.
4. Pick one of those words.
5. To generate the next word, go back to 2.

While this isn't a totally realistic description of the process of writing, it does at least get at the idea that the words in a document are usually topically coherent. More formally, the process above can be described as:

1. For each doc d select a topic distribution $\theta^d \sim Dir(\alpha)$
2. Select a topic $z \sim \theta^d$
3. For each topic select a word distribution $\phi^z \sim Dir(\beta)$
4. Select a word $w \sim \phi^z$

The goal of the LDA learning algorithm then is to maximize the likelihood of our documents, where the likelihood of an individual document d is:

$$p(d|\alpha, \beta) = \prod_{i=1}^{N} p(w_i|\alpha, \beta) \tag{3.1}$$

Marginalizing over the hidden variables z, θ and ϕ we get the following intractable integral.

$$p(w_i|\alpha, \beta) = \int\int \sum_z p(w_i|z, \phi)p(z|\theta)p(\theta|\alpha)p(\phi|\beta))d\theta d\phi \tag{3.2}$$

The integral can be approximated in a few different ways, but we use Gibbs sampling as it has been widely used and often gives more accurate topics.

Gibbs sampling starts by randomly assigning topics to all words in the corpus. Then the word-topic distributions and document-topic distributions are estimated using the following equations:

$$P(z_i|z_{i-}, w_i, d_i, w_{i-}, d_{i-}, \alpha, \beta) = \frac{\phi_{ij}\theta_{jd}}{\sum_{t=1}^{T}\phi_{it}\theta_{td}} \tag{3.3}$$

$$\phi_{ij} = \frac{C_{word_{ij}} + \beta}{\sum_{k=1}^{W} C_{word_{kj}} + W\beta} \tag{3.4}$$

$$\theta_{jd} = \frac{C_{doc_{dj}} + \alpha}{\sum_{k=1}^{T} C_{doc_{dk}} + T\alpha} \tag{3.5}$$

$C_{word_{ij}}$ is the number of times word i was assigned topic j, $C_{doc_{dj}}$ is the number of times topic j appears in document d, W is the total number of unique words in the corpus, and T is the number of topics requested. In essence, the equations above mean that we count the number of times that a word is assigned a topic and the number of times a topic appears in a document, and we use these numbers to estimate word-topic and document-topic probabilities. Once topics have been assigned and distributions have been calculated, Gibbs sampling repeats the process, this time selecting a new topic for each word by looking at the calculated probabilities. The process is repeated until the distributions become stable or a set number of iterations is reached.

We trained LDA topic models on the same 50 million words of English and Urdu data from the BBC as was used to train the Brown language models. We treated each sentence in the data as an LDA "document", and trained two 1000 topic models[4], one for English and one for Urdu. Example topics (the top words associated with each topic) are shown in Table 3.2. Unlike the Brown clusters, LDA clusters are not heavily constrained by grammatical role – verbs, nouns and adjectives can all appear in the same cluster (e.g. *live*, *class*, *poor*). Instead the LDA clusters tend to characterize sets of words that might be used to discuss a particular idea (e.g. *the war on terror* for topic 584).

These topic models were then used to infer topic cluster labels [5] to each word in the Urdu and English training data, to each Urdu word in the test data, and to each word in the BBC English data. As before, the cluster labels on the parallel training data allowed for translation look up, while the cluster labels on the BBC English data allowed a cluster language model to be built[6].

Because the LDA clusters represent shallow semantics, we hoped they would form a new source of information and improve upon the gains we saw with the Brown clusters. Unfortunately, this was not the case and we failed to see any improvement using the LDA clusters. One potential problem was the granularity of the clusters. We ran experiments with both 100 and 1000 topics, but time prevented us from running others. Further investigation would be helpful to understand what prevented the LDA clusters from being a beneficial source of information to Cunei.

3.5.2 Dynamic Automatic Clustering – "Synonym" Clusters

The dynamic clustering approach uses a large monolingual source-language corpus to find synonyms – or other words which preserve the translational structure – for rare and unknown words. These synonyms are substituted for the original term during matching and alignment, and are then treated as gaps during decoding. If the original word is unknown in the parallel corpus, the decoder will be forced to insert a

[4]Models were trained for 4000 iterations, skipping the 100 words with the highest document frequency (the top 100 stopwords).

[5]Inference was run with 500 burn-in iterations, followed by 2 Gibbs sampling chains of 250 iterations and taking a sample of the topic distributions every 50 iterations.

[6]Again, an SRILM 4-gram model with interpolation and Witten-Bell discounting

pass-through arc to fill the gap; if the original word is merely rare, it will have the choice of one or more translations of the original word as potential gap fillers. In either case, the structure of the surrounding text will be preserved where it may not have been if the rare/unknown word had interrupted matching.

This approach was inspired by similar work reported by Meaningful Machines (Carbonell et al., 2006) and Gangadharaiah (Ph.D. thesis proposal). The former uses the approach of finding other words which fill the same context to augment target-language language model lookups. The latter uses the bilingual corpus to find substitutions, together with a fixed back-substitution (for words which are not entirely unknown) after decoding completes.

Dynamic clustering is applied only for words which occur fewer than a given number of times in the source half of the parallel corpus, as such words will cause discontinuities where there are no matched phrases spanning the rare or unknown word.

To determine the candidate synonyms for a word, a specific number of words on the left and right of the word to be replaced are taken from the input sentence and form the context. All instances of the left context and all instances of the right context are retrieved from both the source-language corpus and the source half of the parallel training corpus (which is fast since the indices are designed specifically for this operation) and are intersected to determine all locations where the full context occurs. The words within the resulting gaps are accumulated to determine the set of candidate replacements. These candidates are filtered by absolute frequency within the given context and total frequency of occurrence in the parallel corpus – it would not make sense to replace a rare word by an even rarer word, even if it is quite common in the monolingual corpus.

This process is repeated with successively fewer words of context until sufficient replacement words have been found or the minimum acceptable context size is reached. If more than the desired number of replacements are found, they are ranked by context size and both relative and absolute frequencies, and the lowest-ranking replacements are eliminated from consideration.

The synonyms thus found are inserted into the confusion lattice for the input sentence, and the matcher then proceeds to look up all phrases containing either the original word or one of its replacements. Synonyms are marked in the confusion lattice's nodes in such a way that they produce gaps when the decoder assembles the resulting partial translations into a complete translation. Figure 3.7 shows an example of the candidate substitutions found for the word "barred".

As of the end of the 2009 SCALE, synonym substitution had produced only a minor improvement on the development set and an insignificant change on the devtest and test sets. We now believe that two enhancements are likely to produce positive results for the approach: the use of part-of-speech tags to filter candidate replacements and a two-phase parameter tuning regime. Many of the replacements produced by the dynamic clustering look good monolingually, but do not properly preserve the structure of the translation because they have a different part of speech than the original word; by eliminating such mismatches, unwanted divergences will be minimized. In Figure 3.7, "against" is such a mismatched replacement. During the summer, we used a joint optimization of all parameters, but since synonym substitution applies to only a small percentage of the input, it is probably better to first optimize parameters with substitution disabled and then run a separate optimization which fixes the values of all parameters *except* those which control synonyms.

3.6 Conclusions

One of the key design principles of using partial structured models within Cunei is that multiple sources of information can be combined together. So while development during most of the summer focused independently on integrating each new source of information, at the end of SCALE we were able to merge everything into one system. Figure 3.8 shows how the input was annotated with HIVEs, Brown cluster la-

bels, and LDA cluster labels. Feature functions fired independently for each of these sources of information and Cunei's optimization process set weights based on the utility of each information source.

It should also be pointed out that over the course of the summer bugs were fixed and minor improvements were made to various components within Cunei. In addition to using the new sources of semantic and syntactic information, the final system also combines multiple alignments, models sentential context, and models empty translations. However, comparatively little time was spent on these components since they were not the focus of the workshop.

The results of these efforts are shown in Figure 3.9. We are proud to show significant improvement over our baseline in just a few short weeks of collaboration. This is highly encouraging and demonstrates the usefulness of semantic and syntactic meta-information. In further work it would be helpful to investigate which sources of information provide the most 'bang for the buck'. In addition, while using a larger corpus and using multiple sources of meta-information are not mutually exclusive, it would be interesting to compare the relative cost of these two tasks with their improvement in translation quality.

```
<?xml version="1.0" encoding="UTF-8"?>
<output system="cunei">
<segment id="1">
  <hypothesis id="0" score="-84.4914">
    <source>
      <span id="0">
        <sequence type="Lexical">اور مشرف</sequence>
        <sequence type="Brown-Clusters">110 7</sequence>
        <annotation start="0" end="1" type="ENAMEX">PERSON</annotation>
      </span>
      <span id="1">
        <sequence type="Lexical">نظیر بے</sequence>
        <sequence type="Brown-Clusters">175 527</sequence>
        <annotation start="0" end="1" type="ENAMEX">PERSON</annotation>
        <annotation start="0" end="2" type="ENAMEX">PERSON</annotation>
        <annotation start="1" end="2" type="ENAMEX">PERSON</annotation></span>
      <span id="2">
        <sequence type="Lexical">ملاقات کی</sequence>
        <sequence type="Brown-Clusters">4 646</sequence>
      </span>
    </source>
    <target>
      <span id="0">
        <sequence type="Lexical">musharraf and</sequence>
        <sequence type="Brown-Clusters">756 7</sequence>
        <annotation start="0" end="1" type="ENAMEX">PERSON</annotation>
      </span>
      <span id="1">
        <sequence type="Lexical">benazir bhutto</sequence>
        <sequence type="Brown-Clusters">714 436</sequence>
        <annotation start="0" end="1" type="ENAMEX">PERSON</annotation>
        <annotation start="0" end="2" type="ENAMEX">PERSON</annotation>
        <annotation start="1" end="2" type="ENAMEX">PERSON</annotation></span>
      <span id="2">
        <sequence type="Lexical">met</sequence>
        <sequence type="Brown-Clusters">824</sequence>
      </span>
    </target>
    <alignments> ... </alignment>
    <features> ... </features>
  </hypothesis>
```

Figure 3.4: XML Output

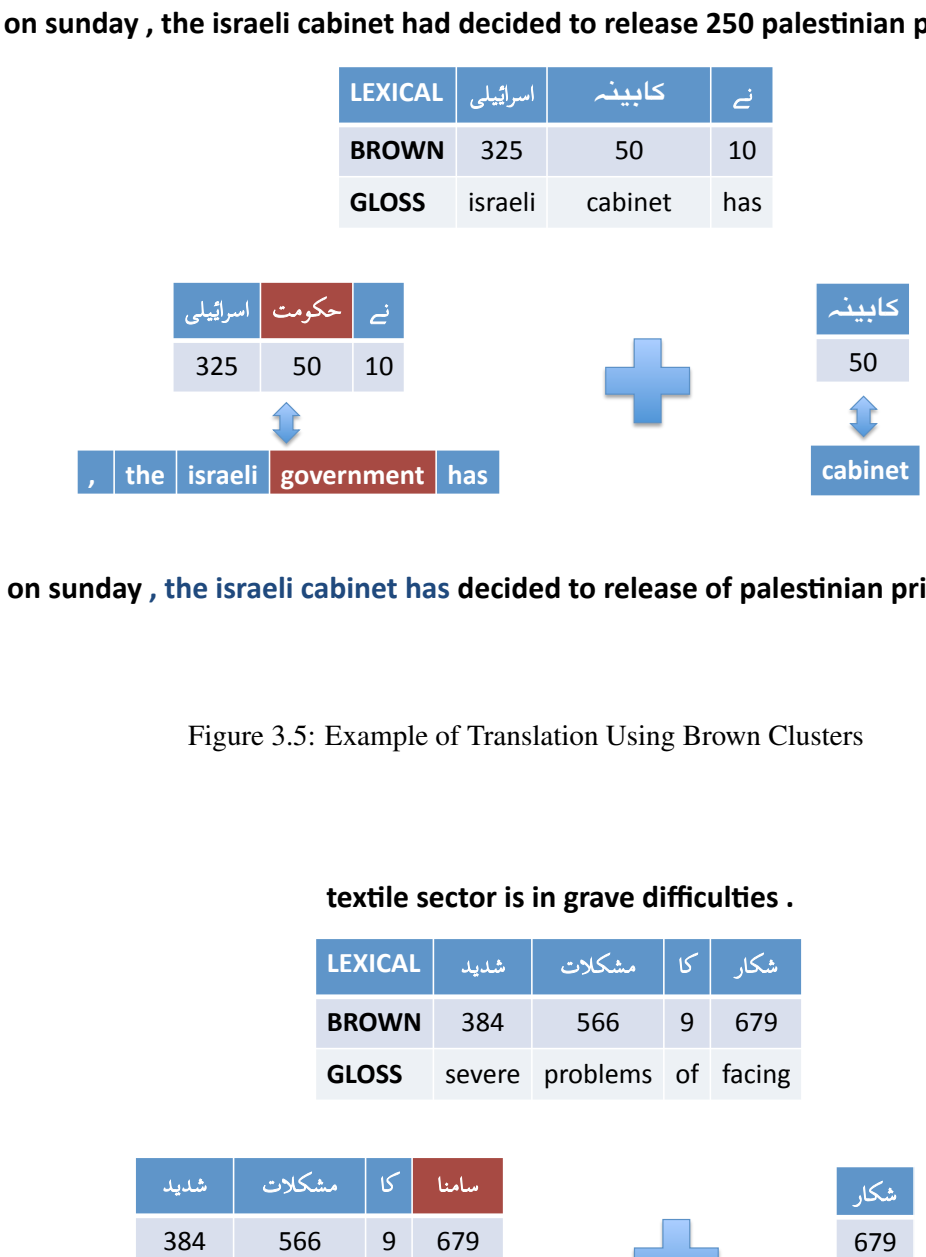


Figure 3.5: Example of Translation Using Brown Clusters

Figure 3.6: Example of Translation Using Brown Clusters

israeli	security	forces	barred	palestinians	under	45	from	praying
			against					
			said the					
			fighting					

Figure 3.7: Example of Synonym Clusters

LEXICAL	مشرف	اور	بے	نظیر	کی	ملاقات
BROWN	110	7	175	527	4	646
LDA	43	اور	33	11	کی	20
ANNOTATION	PERSON		PERSON			

Figure 3.8: Combining Multiple Types of Meta-Information

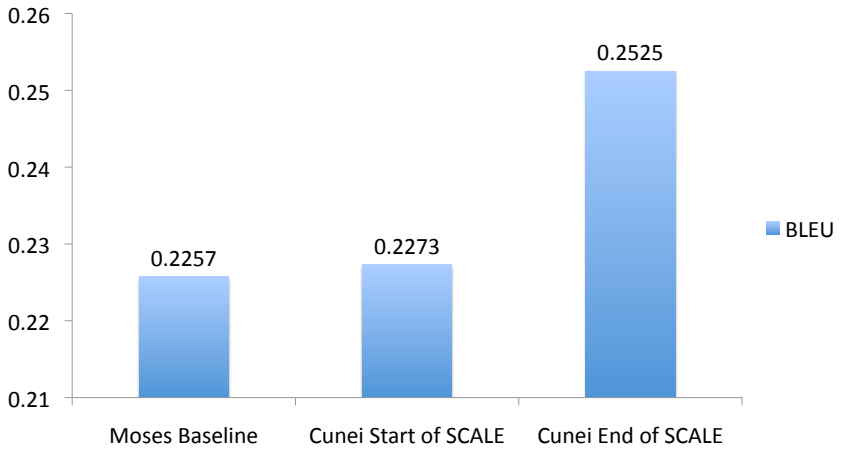


Figure 3.9: Evaluation of Cunei

Chapter 4

Finding Parallel Sentences in Comparable Corpora

Finding parallel sentences is always a problem for machine translation, particularly for low density languages such as Urdu. This section describes how we acquire parallel sentences automatically, using large monolingual corpora of Urdu and English collected from the BBC. We use an existing machine translation system to translate the Urdu corpus into English sentences, and then apply an information retrieval system to select sentences from the English corpus that look similar. We then train a support vector machine to classify retrieved sentences as translations or not, achieving 94.0% accuracy, a 49% error reduction over the baseline. The sentences classified as translations can then be used as additional training data for the machine translation model.

A major factor limiting the performance of most machine translation systems is the availability of large numbers of parallel sentences for training. This lack of such resources is especially problematic for low density languages such as Urdu, where often fewer than 100,000 parallel sentences are available, while machine translation systems for high density languages typically have 1,000,000 parallel sentences or more.

Traditionally, the parallel sentences used as training data come from paid translators and are relatively expensive and time consuming to construct. Recent work has shown that parallel sentences can sometimes be acquired from comparable corpora, by using a variety of heuristics to identify sentences that are likely to be translations of each other (Yang and Li, 2003; Utiyama and Isahara, 2003; Munteanu and Marcu, 2005; Wu and Fung, 2005; Abdul-Rauf and Schwenk, 2009). We build upon this work, introducing a novel approach to selecting likely translations that combines an information retrieval system for proposing English sentences similar to machine translation output, with support vector machine classifiers which use features such as sentence length and the BLEU score to classify the retrieved sentences as actual translations or not.

4.1 Related Work

Yang and Li used sequence alignment techniques to recover parallel English and Chinese titles from partially parallel Hong Kong corpora (Yang and Li, 2003). They first translate the words in the English titles into Chinese characters using a dictionary, and then search the Chinese titles for the one with the longest common substring match. On corpora of Hong Kong government and banking text where only some documents are published in both languages, they are able to recover the parallel titles with F-measures of 89-97.

Utiyama and Isahara applied cross-language information retrieval (CLIR) and alignment through dynamic programming to find parallel sentences in a partially parallel Japanese and English newspaper corpus (Utiyama and Isahara, 2003). Their CLIR component glossed Japanese articles with English words using a bilingual dictionary, and used information retrieval style bag-of-words similarity to identify similar Japanese

and English articles. They then took all pairs of sentences in the (assumed) parallel articles and produced alignment scores using dynamic programming. The aligned sentences were ranked based on a combination of both their sentence and document alignment scores, and this ranked list produced a precision of 98% for the top 150,000 sentences.

Munteanu and Marcu used CLIR and a maximum entropy classifier to find parallel sentences in comparable corpora from English, Chinese and Arabic Gigaword and bilingual news websites (Munteanu and Marcu, 2005). They identified parallel articles using CLIR like Utiyama and Isahara, keeping only article pairs published within a 5 day window of each other. They then took all pairs of sentences in the parallel articles, filtered them based on word overlap, and trained a maxiumum entropy classifier using features like the lengths of the sentences and the number of words left unaligned by a statistical alignment model. When providing these parallel sentences to a machine translation model, they saw no effect when training on their full 95 million word parallel corpus, but did see 5-10 point improvements in BLEU for parallel corpora of 1 million words or smaller.

Wu and Fung combined CLIR with inversion transduction grammars to find parallel sentences in comparable corpora from the Chinese and English Topic Detection and Tracking (TDT) data (Wu and Fung, 2005). They identify parallel articles using CLIR as in previous work, repeating the CLIR process at the sentence level to select likely translations from all pairs of sentences in their parallel articles, and then used an iterative EM-based algorithm to expand the sets of parallel articles and sentences. They scored the candidate parallel sentences using inversion transduction grammars, and reported an average precision of 65%.

Abdulrauf and Schwenk combined a baseline machine translation (MT) system with mononlingual information retrieval and a translation error rate (TER) filter to identify parallel sentences in comparable corpora from the English and French Gigaword corpora (Abdul-Rauf and Schwenk, 2009). A baseline MT system was used to translate all French sentences into English, and information retrieval was used to select similar sentences from the English corpus, restricting to those published within a 5 day window of the original French sentence. Sentence pairs with low translation error rate (TER) were filtered, and unmatched words were stripped based on word error rate (WER). When providing the resulting sentence pairs as additional training data for an MT system, they found 2 point improvements in BLEU when training on 2 million words of parallel sentences, but only 0.2 point improvements when training on 40 million words.

Our system is most like that of Abdulrauf and Schwenk in that we combine a baseline machine translation system with information retrieval techniques to select candidate parallel sentence pairs. However, instead of relying on heuristics like a 5 day window or a TER threshold, we provide such bits of information as features to a support vector machine classifier, and allow the classifier to determine the most appropriate weighting of the features.

4.2 Monolingual Data

We collected news articles from the BBC website for both English and Urdu up through February 2009. Since much more English data was available than Urdu data, we selected a subset of English articles more likely to match the Urdu articles. More than 85% of all Urdu articles were categorized by BBC under *pakistan*, *regional*, *specials*, *india*, *sport* or *news* so for English articles, we used only the sections *europe*, *americas*, *asia-pacific*, *africa*, *south_asia*, *middle_east*, *sport*, *in_depth* and *special_report* and discarded sections like *uk*, *health* and *business* which seemed to be absent from the Urdu side.

The resulting corpora, after various automatic cleanups to remove headers and menus and normalize tokenization, are shown in Table 4.1. About 2 million English sentences and 1 million Urdu sentences were collected, corresponding to 50 million words and 30 million words, respectively. These corpora formed the basis for our experiments.

	Words	Sentences
English	54,873,439	2,250,108
Urdu	28,530,184	953,524

Table 4.1: The BBC English and Urdu data.

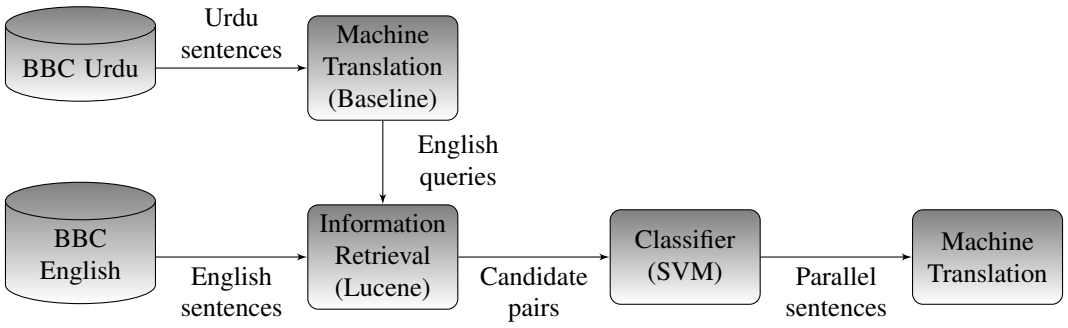


Figure 4.1: Retrieving parallel sentences.

4.3 Sentence Selection Model

Figure 4.1 gives an overview of the system. First, Urdu sentences are translated into English using a baseline machine translation (MT) system, in our case, Moses trained on the existing Urdu-English parallel data from NIST. The English MT output sentences are then used as queries to a Lucene index of the BBC English data, and similar English sentences are retrieved for each MT output query.

At this point, for each Urdu sentence we have a single English MT output sentence and several English sentences retrieved from the BBC corpus. Each retrieved sentence is passed along with its corresponding Urdu sentence and English MT output sentence to a support vector machine classifier. The classifier then predicts whether or not the retrieved sentence is a translation of the original Urdu sentence. All sentences classified as true translations are paired with their Urdu sentences and provided as additional training data for the machine translation system.

Of course, for this scheme to work, we must be able to train the classifier on retrieved English sentences (and their corresponding Urdu sentences and English MT output) for which we know ahead of time whether they are translations or not. However, the BBC data is unlabeled and we don't know which sententences are translations of each other. One solution to this training data problem would be to hand annotate some of the retrievals. This could even be done by English monolinguals by just comparing the English BBC sentence to the English MT output sentence. For example the following sentences appear to be translations of each other:

MT output israeli prime minister موسسہ، مگسک موالسہ ہو said that the rocket attacks on israel , it was clear that the resolution is " intolerable process .

BBC retrieval israeli prime minister ehud olmert said the continued rocket attacks on israel showed the resolution was " unworkable " .

Another solution to the training data problem, and the one we adopt here, is to take a few pairs of known Urdu-English sentence translations, add the English halves to the Lucene index, and use the Urdu halves to generate queries. Then, for each sentence output by the retrieval system, we label it $+1$ if it matches

the known English translation or -1 if it doesn't. Generally, we expect that the retrieved sentences that are actually the known English translations and are labeled $+1$ will be at the top of the retrieval list, and the remaining sentences retrieved from the BBC will not be translations[1].

To build our training data then, we took our NIST development data as the known Urdu-English translations. This data included 981 Urdu sentences with around 4 English translations each, and we retrieved 20 English sentences for each Urdu sentence. The result was 18507 training instances, with 2179 positive examples and 16328 negative examples. Note that only about 13% of the data were positive examples though we might have expected $4/20 = 20\%$. The drop is in part due to translations that were not retrieved, but also because about 10% of the English translations were exact duplicates of each other, and therefore discarded for training purposes.

We extracted features for each training instance (Urdu sentence, English MT output and English retrieved sentence) looking at important factors like sentence length, presence of numeric words, and the scores generated by various MT evaluation metrics. We generated the following features:

- The ratio of the number of words in the retrieved BBC sentence to the number of words in the MT output sentence.
- The difference between the number of words in the retrieved BBC sentence and the number of words in the MT output sentence.
- A binary feature indicating whether or not the retrieved BBC sentence and the MT output sentence had exactly the same number of numeric words.
- The ratio of the number of numeric words present in both the retrieved BBC and MT output sentences to the number of numeric words present in either.
- The BLEU metric comparing the BBC and MT output sentences.
- The NIST metric comparing the BBC and MT output sentences.
- The TER metric comparing the BBC and MT output sentences.
- The TERP metric comparing the BBC and MT output sentences.
- The METEOR metric comparing the BBC and MT output sentences.

These feature vectors, paired with their positive and negative labels, formed the input for training our classifier.

4.4 Experiments

We used the SVM-perf implementation of support vector machines (SVMs) (Joachims, 2005) to train classifiers from our data. For evaluation purposes, we ran a 10-fold cross-validation, setting SVM hyperparameters using a grid search and 9-fold cross-validations on just the training folds. This approach can produce different parameters for each fold[2] but avoids artificially inflating our estimated performance. The 10-fold

[1]Note that this process can generate slightly noisy training data if the BBC data contains real translations of our known Urdu sentences that don't match the known English sentences. However, in preliminary experiments this didn't seem to be a common problem.

[2]Though in practice, the error rate loss function was always selected, and in 8 of 10 folds, the cost of misclassification was set to 100.

cross-validation achieved a classification accuracy of 94.0%, a 49% error reduction over the 88.2% accuracy achieved by a majority class (all negative) baseline model.

Using the parameters selected by the 10-fold cross-validation[3], we trained a model using the entire training data, and have begun to use the resulting classifier in the full pipeline illustrated in Figure 4.1. Output generally looks promising – the *israeli prime minister* example above was selected by the system, as was the following example which appears to be a correctly identified parallel sentence pair:

MT output he ںلگ ےہ of one of the buildings in the old city , jordan ، وہ ںکلک ملگ britain and the لس ق وف ںکلک ےق wall of china .

BBC retrieval they include rome 's colosseum , jordan 's ancient city of petra , britain 's stonehenge and the great wall of china .

However, we also see output sentences which suggest the need for additional features. For example:

MT output in response to a question , he said that the ےہ ںہ وہ ملک اسے action against india or any other country , but not to the united nations has demanded .

BBC retrieval but the yugoslav ambassador to the united nations , vladislav jovanovic , said that action against his country would mark nato as an aggressor .

Here, the sentences are clearly not translations because there is substantial mismatch between the named entities of the sentences. Adding features to count matching named entities would likely alleviate this problem. We also imagine that including date based features – e.g. were the containing articles published within 5 days of each other? – would address some of these issues as well.

4.5 Discussion

While the performance of the SVM is encouraging, the data was generated by artificially inserting sentences with known translations into the index, and so the resulting performance may not fully reflect the performance to be expected on the BBC data alone. Thus one of the next steps is to evaluate the performance of the clasifier on the BBC data alone, using the help of human annotators. We plan to do this using Amazon's Mechanical Turk, which should be a cheap and efficient way of quickly determining the accuracy of our system in finding parallel sentences. As suggested earlier, this work can probably be done by English monolinguals, by just comparing the MT output sentences to the BBC retrieved sentences. And any data on which translations were right or wrong can be added as additional training data for the classifier.

Of course, the real test of our parallel sentence identification system is how much the additional sentences improve the performance of a machine translation system. Prior work suggests that parallel sentences derived from comparable corpora often help MT when only 1 or 2 million words of parallel text are available. Since this is the case for the Urdu-English pair, and since preliminary experiments indicated that parallel sentences might be found for 1-3% of our 1 million BBC Urdu sentences, this appears to be a promising avenue of continued work.

[3] the error rate loss function and a cost of misclassification of 100

Chapter 5

Inducing Translations from Monolingual Texts

Current state of the art machine translations systems, (Li et al., 2009), (Phillips, 2009), require parallel corpora to produce translations of passable quality by any standard. Creating these texts is arduous, requiring both the availability of bilingual speakers (for long periods of time) and the cost thereof. We aim to move away from this paradigm, examining the resources easily acquirable: a number of small seed lexicons[1], a massive amount of monolingual corpora (available from the internet) in any number of languages, and adequate[2] compute resources. The goal is to bootstrap larger bilingual lexicons from the small bilingual dictionaries using the large monolingual texts.

Previously attempts have been made to induce bilingual lexicons from monolingual texts (Rapp, 1995; Fung and Yee, 1998; Rapp, 1999; Koehn and Knight, 2000; Mann and Yarowsky, 2001; Koehn and Knight, 2002; Schafer and Yarowsky, 2002a; Haghighi et al., 2008; Garera et al., 2009). The work presented here differs from past work because they only explore the use of a pair of languages, and they generally applied their work to language pairs which are high resource in reality. Here we examine going beyond language pairs to language triples (or to language tuples of arbitrary size), and explore the scenario where next to no parallel text is available (so called *low resource* or *less commonly taught languages*). In low resource conditions, many linguistic tools which depend on parallel corpora for their creation are not available (e.g. syntactic parsers, lemmatizers, part of speech taggers). Our goal is to create high-precision multi-lingual lexicons, ultimately for inclusion in machine translation systems.

Our research takes into account previous work on bilingual lexicon induction, bridge languages, pivot languages, and a healthy dose of statistical pattern recognition. We hope to avoid the need for combinatorics, and aim to keep their explosion from having any impact upon our processes.

Our assumptions follow:

- The seed lexicons used (which are primarily induced from parallel text or via some other automatic and emprical method) are not necessarily high-precision.
- Using multiple languages will allow for a "triangulation" of sorts, where the noise in the matching between a "target" and single "foreign" language will be noisier than one that must match two or more "foreign" languages.

[1] Where the definition of "small" is relative to the desired lexicon size.

[2] Where "adequate" is hopefully little more than a high-performance desktop.

5.1 Semantic Tunneling Framework

The general ideas behind Semantic Tunneling[3] sketched in Section 1 are made concrete in this section with the presentation of a theoretical framework. Additional details about specific components of our framework, in addition to experiments and results, are given in following sections as indicated.

5.1.1 Notation

A generic language is denoted $\mathbb{L}$ and a specific language is denoted, e.g., $\mathbb{E}$ for English. Thus, if the set of languages of interest include $\mathbb{E}$ English, $\mathbb{F}$ French, $\mathbb{G}$ German, $\mathbb{A}$ Arabic, $\mathbb{H}$ Hindi, $\mathbb{S}$ Spanish, $\mathbb{U}$ Urdu, and $\mathbb{Z}$ Zulu, then $\mathbb{L} \in \{\mathbb{E}, \mathbb{F}, \mathbb{G}, \mathbb{A}, \mathbb{H}, \mathbb{S}, \mathbb{U}, \mathbb{Z}\}$, and $\mathbb{L}_i$ denotes a word in language $\mathbb{L}$.

There are many ways to summarize word-usage and context or semantic information about a collection of documents in a language. For example, if there are $\mathrm{N}_{\mathbb{L}}$ unique iotas or terms in language $\mathbb{L}$, and $\mathrm{m}_{\mathbb{L}}$ iotas in the lexicon for language $\mathbb{L}$, where $\mathrm{m}_{\mathbb{L}} \leq \mathrm{N}_{\mathbb{L}}$, then a word co-occurrence matrix is an $\mathrm{m}_{\mathbb{L}} \times \mathrm{m}_{\mathbb{L}}$ matrix of word co-occurrences for the collection of documents in language $\mathbb{L}$ (see Section 5.2.4 for more details).

This word co-occurrence matrix represents one *view* of the data for a given language. Other views that we identified as being of interest, such as date-distribution similarity and string edit distance, are further discussed in Section 5.2.4. In general, we denote $\mathrm{S}^i_{\mathbb{L}}$ as view i for language $\mathbb{L}$.

For $\mathrm{m}_{\mathbb{L}_1}$ iotas in language $\mathbb{L}_1$, and $\mathrm{m}_{\mathbb{L}_2}$ iotas in language $\mathbb{L}_2$, using, for example, available hand-curated dictionaries or alignment induced dictionaries, a $\mathrm{m}_{\mathbb{L}_1} \times \mathrm{m}_{\mathbb{L}_2}$ matrix $\mathrm{L}_{\mathbb{L}_1\mathbb{L}_2}$ of translation probabilities can be constructed, and $\mathrm{L}_{\mathbb{L}_1\mathbb{L}_2}$ is called a *link matrix*. That is, entry (i, j) of link matrix $\mathrm{L}_{\mathbb{L}_1\mathbb{L}_2}$ records the probability that iota i in $\mathbb{L}_1$ is translated as iota j in $\mathbb{L}_2$.

5.1.2 Framework

The above concepts of views, lexicons, and link matrices is illustrated in Figure 5.1, which presents a Semantic Tunneling framework. For all the languages of interest, the S matrices for view i are arranged along the diagonal of a larger matrix, with the link matrices on appropriate off-diagonal positions. This arranged multi-language matrix is known as a *slice*, and allows for information from multiple source languages to be utilized, such as via triangulation, to enhance target word translation.

To be clear, note that a given slice contains the same view of the data (e.g., word co-occurrence) for a set of languages of interest, and a different slice contains different views of the data, but for the same set of languages. For example, in Figure 5.1, $\mathrm{S}^1_{\mathbb{E}}$ in slice 1 is one view of the English dataset and $\mathrm{S}^2_{\mathbb{E}}$ in slice 2 is a second view. Moreover, the link matrices are the same across all slices. In other words, in Figure 5.1 the $\mathrm{L}_{\mathbb{E}\mathbb{F}}$ link matrix in slice 1 is the same $\mathrm{L}_{\mathbb{E}\mathbb{F}}$ link matrix in slice 2.

In addition, our framework allows for multiple slices to be combined, or *fused*, to also enhance target word translation, since different views of the data provide different qualities of information, and it is believed that the fusion of multiple pieces of information can be of higher quality than using only one piece of information for a translation task. See Section 5.3.2 for more detail on fusion. Finally, we note that the matrix slices, which constitute a high-dimensional space, can be *embedded* into a lower-dimensional space, for the possibility of enhanced processing, especially because of noise reduction. See Section 5.3.3 for more detail on embedding.

Finally, we note that as new words are learned, our views and link matrices can be grown over time by adding appropriate rows and columns. This strategy is especially useful in the scenario where high-quality *seed* words are used as the initial iotas in the views and link matrices. Further details about how the slices can be constructed and used are presented in Section 5.4.

[3] We beg off explanation of the phrase *Semantic Tunneling* until the very end.

$$\begin{bmatrix} \mathrm{S}^1_{\mathbb{E}} & & & \\ \mathrm{L}_{\mathbb{EF}} & \mathrm{S}^1_{\mathbb{F}} & & \\ \vdots & \vdots & \ddots & \\ \mathrm{L}_{\mathbb{EU}} & \mathrm{L}_{\mathbb{FU}} & \dots & \mathrm{S}^1_{\mathbb{U}} \end{bmatrix} + \begin{bmatrix} \mathrm{S}^2_{\mathbb{E}} & & & \\ \mathrm{L}_{\mathbb{EF}} & \mathrm{S}^2_{\mathbb{F}} & & \\ \vdots & \vdots & \ddots & \\ \mathrm{L}_{\mathbb{EU}} & \mathrm{L}_{\mathbb{FU}} & \dots & \mathrm{S}^2_{\mathbb{U}} \end{bmatrix} \dots$$

Figure 5.1: Semantic Tunneling Framework.

5.2 Data

5.2.1 Corpus Collection and Creation

Parallel Corpora

We the English, Spanish, and French pieces of the proceedings of the European Parliament (Europarl) for our parallel corpora (Koehn, 2005). For an English-Urdu parallel corpus, we used the corpus for the NIST 2008 evaluation.

Monolingual Corpora

Our monolingual corpora was taken from a web crawl of the BBC news (http://www.bbc.co.uk). In these experiments we used news stories from English, French, Spanish, and Urdu. Each language had approximately 30 thousand news stories. Since these news stories are from the same source and cover the same date range, we assume that these corpora are comparable.

Preprocessing

In keeping with the spirit of our challenge, we did minimal preprocessing, only applying a technique f if it was possible to create with little to no specific linguistic knowledge of the target language. We used the internal Mac OS X utility *textutil* to strip the documents of HTML tags. We normalized using an HLTCOE internal script produced by Jim Mayfield (and contributed to by Aaron Phillips, Gramm Richardson, and Eric Case) and split the data into tokens using scripts from Cunei (Phillips, 2009). Some of the peculiarities of the Urdu language encoding and equivalent characters were captured by the normalization script, but in no way did we perform stemming or lemmatization on the data, as many previous authors have done.

5.2.2 Dictionary Creation

For most of our experiments we used hand-curated dictionaries graciously provided by Charles Schaeffer and David Yarowsky. For details on the creation of these dictionaries, see (Phillips, 2009). The alignment-induced dictionaries were provided by Steven Bethard, produced from the English-Urdu parallel corpus by symmetrized alignments using various aligners (e.g. Giza++, Basis aligner), discarding word pairs for which all aligners did not agree.

5.2.3 Tuple Creation

We sought m seeds by symmetrizing the conditional probabilities[4] for terms translated from $i \rightarrow j$ and $j \rightarrow i$ where $i, j \in \mathbb{L}$. The m seeds are the iotas where the symmetrization for all $\mathbb{L} \times \mathbb{L}$ pairings match.

English French Tuples

Symmetrization for the lexicons induced from the Europarl parallel corpus progressed as follows: A lexicon exists populated with probabilities ($p = [0, 1]$) that a term in $\mathbb{L}_1$ is translated as a term in $\mathbb{L}_2$, we denote this lexicon as $\mathbb{L}_1 \rightarrow \mathbb{L}_2$, and the probability that i is translated as j $p(i|j)$.

For every pair of languages in $\mathbb{L} = \{\mathbb{E}, \mathbb{F}, \mathbb{S}\}$ exists a translation lexicon derived from the Europarl corpus. Thus, we are given $\mathbb{L}_1 \rightarrow \mathbb{L}_2$ and $\mathbb{L}_2 \rightarrow \mathbb{L}_1$. Since we sought a high-precision lexicon for use in our proof-of-concept experiments, we applied a threshold (τ) to the lexical entries, only accepting terms where $p(i|j) > \tau$. Further, we excluded from consideration any pair (i, j) that did not have the property $p(i|j) > \tau$ and $p(j|i) > \tau$. In essence, this excluded terms which are not reflected in $\mathbb{L}_1 \rightarrow \mathbb{L}_2$ and $\mathbb{L}_2 \rightarrow \mathbb{L}_1$. This yields a symmetrized high-precision lexicon, denoted by $\mathbb{L}_1 \leftrightarrow \mathbb{L}_2$ or $T^{\mathbb{L}_1\mathbb{L}_2}$.

English, Spanish, French Tuples

Similar to the English-French Tuples above, we created two sets of tuples for the English, Spanish and French languages. The first, a stricter set, was created by applying another level of symmetrization to the existing lexical entries from $T^{\mathbb{EF}}$, by excluding all of the lexical items for which a mapping was not present in all pairs of languages ($\mathbb{E} \leftrightarrow \mathbb{F}$, $\mathbb{E} \leftrightarrow \mathbb{S}$, $\mathbb{F} \leftrightarrow \mathbb{S}$), thus yielding $T^{\mathbb{ESF}}_{rigid}$.

These lexical terms are almost entirely person names, place names, and geo-politcal entities (from various regions of the world). We conjecture this is an effect of the corpus used to induce these lexicons, but it cannot be ruled out as an effect of this technique applied to any trilingual corpus. If we can reproduce this list of *rigid designators* in other languages (specifically $\mathbb{U}$), this may fulfill a number of needs: transliteration training data and search features for discovering bi- or tri-lingual sentences immediately come to mind.

English Urdu Tuples

Two sets of English-Urdu tuples were created: $T^{\mathbb{EU}}_{align}$, derived from alignment-induced dictionaries and $T^{\mathbb{EU}}_{hand}$, derived from hand-curated dictionaries. $T^{\mathbb{EU}}_{align}$ was created in a similar fashion to $T^{\mathbb{EF}}$, since it is induced from alignments on parallel corpora. $T^{\mathbb{EU}}_{hand}$, on the other hand, was created in a slightly different fashion. The set intersection of three hand-curated English-Urdu dictionaries formed $T^{\mathbb{EU}}_{hand}$.

5.2.4 S-Matrices

As discussed in Section 5.1.1, the S-matrices summarize the documents for a given language using a particular view of the data, such as word co-occurrence, data-distribution similarity, and string-edit distance. The current focus was utilizing word co-occurrence views of the data.

Word Co-Occurrence

For words i and j, if $\{d : i \in d\}$ is the set of documents containing word i, then the co-occurrence of j with i is given by the number of times word j occurs in the documents containing word i:

$$\mathrm{S}_{ij} = |j \in \{d : i \in d\}|$$

[4] Conditional probabilities provided courtesy of Chris Callison-Burch

While this is a natural metric to characterize words j that occur often with word i, thereby measuring a contextual relationship between i and j, co-occurrence scores can be biased larger simply due to the fact that longer documents have more words. One way to help dampen this problem is to normalize the co-occurrence scores:

$$\text{Normalized } \mathrm{S}_{ij} = \frac{\mathrm{S}_{ij}}{\#j}$$

by dividing S_{ij} by the total number of words j in the corpus, denoted by $\#j$.

This can be thought of as taking measurements from the area around the iota in question, or capturing co-occurrence in this document-sized window. One can imagine that defining *window*[5] differently would capture different granularities of co-occurrence. We define a *5-window* as the iota in question (in the middle) and two iotas to either side. Likewise a *9-window* is the iota in question and four to either side. Since these capture different types of co-occurrence (5 for within the same phrase, 9 for something more like a sentence), we argue in favor of using them in concert, rather than using any single one (as described in Section 5.3.2).

One problem with S_{ij} is that low-information j words, say j's that occur in every document, will result in high S_{ij} values. Instead we would like high S_{ij} scores to result when j's co-occurrence is meaningful in the context of i. So one idea is to somehow down-weight the noisy high S_{ij} scores that result from low-information j words.

In the document-vector representation of a corpus, as opposed to a co-occurrence representation, this concept is captured by the TF-IDF score (term frequency, inverse document frequency). If $|D|$ is the total number of documents in the corpus, and if $|d : t_i \in d|$ is the number of documents where the term t_i appears, assuming $n_{ij} \neq 0$[6]. For term i and document j:

$$TF_{ij} = n_{ij},$$

or the number of times term i occurs in document j, and

$$IDF_i = \log \frac{|D|}{|d : t_i \in d|}.$$

This gives:

$$TFIDF_{ij} = TF_{ij} \times IDF_i.$$

The TFIDF weights are large for high-frequency words *and* words that do not occur in a large number of documents relative to the size of the corpus. This effect helps to filter out common, low-information terms.

In order to adapt this down-weighting concept to word co-occurrence matrices, we construct a TF-IDF-weighted co-occurrence matrix instead of a normal word co-occurrence matrix by summing the TF-IDF-weighted document-vector terms instead of raw frequency counts.

- TF_{ij}: frequency of word j in the context of word i. This is S_{ij}. Larger numbers indicate that j occurs often when i occurs. With the normalized TF_{ij}, a large value (close to 1) indicates that j occurs often when i occurs, and only when i occurs.
- IDF_i: a down-weighting for high-frequency words that carry little information.

[5] *Beam* and *window* are used interchangeably to refer to these measures.

[6] Some use $1 + |d : t_i \in d|$ to prevent this division by 0 problem.

Rigid Designators

Cities of the world, famous people, geo-political entities, and other widely used proper nouns often occur as transliterations in each language[7]. Thus, their phonetic similarity maps to similar (and at times identical) orthographies. More generally, a class of transformation and transliteration rules can be quickly designed (as is evidenced by Chapter 7). Being able to identify the mapping of such *rigid designators* across multiple languages will serve as good anchors and links for context. In fact, such may work as a crude topic-classification, since "Obama" or "Clinton" are likely to be referenced in an article having to do with politics, while "Dalai Lama" may be referenced in articles with a more religious bend.

With this information, we could create a vector of co-occurrences of each iota in a language with each rigid designator (e.g. $v_i = 1$ if the iota occurs in the same document as rigid designator i and 0 otherwise) . The distance between a pair of iotas in this dimension can be determined by applying the distance measures (described further in Section 5.3.1) to the pair of vectors.

Date Distribution Similarity

International-scale events show up in the same news stories within a 24 hour range of each other, and thus can give us an additional contextual clue nearly orthogonal to iota co-occurrence. Specifically, "Swine" (from the first "Swine Flu" articles, which we were fortunate to capture the first news stories of in our corpus) and "Earthquake" (Schafer and Yarowsky, 2002b) are prime examples of iotas whose time signatures are strongly correlated with widespread world events– events that are reported in most, if not all, languages. So, similar time-signatures for words across languages may be evidence in favor of a match. This deserves further investigation, since some patterns are likely more telling than others: iotas whose signatures are sporadic and spiky are likely more useful than those that are ever-present and flat.

A distance between two time-signatures is likely not as straightforward as that described for rigid designators, when accounting for time zone differences and geographic spread. There is a plethora of literature on pattern recognition in signals that deals with a related problem, so we will turn there for solutions.

5.2.5 L-Matrices

The link matrices (or L, the $\mathrm{m}_{\mathbb{L}_i} \times \mathrm{m}_{\mathbb{L}_j}$ matrices mentioned in Section 5.1.1) are the off-diagonal blocks between two S matrices. Though they take on many forms through the experimentation, they always represent translation probabilities between the iotas in the S matrices connected by the L matrix.

The simplest case (from which we draw our results reported in Section 5.3.3), is when $|\mathrm{m}_{\mathbb{L}_i}| = |\mathrm{m}_{\mathbb{L}_j}|$, and there is a one-to-one mapping between the iotas in $\mathbb{L}_i$ and $\mathbb{L}_j$. In this case, the Ss are ordered so the rows and columns of S_i correspond to the rows and columns of S_j, thus L_{ij} is the identity matrix. Equivalently, when working with only monosemous words, L may be a reordered identity matrix.

When m contains polysemous words, we employed three methods. First, we force monosemy by composing L as above. Second, for all i and j in L: $\mathrm{L}(ij)$ is 1 if i is a translation of j, and 0 otherwise. Third, same as the second, but the rows are normalized to sum to 1. Results in Section 5.5 were derived using the second and third approaches outlined. Specifically, Figure 5.4 compares these two approaches to contending with polysemy.

5.3 Experimental Design

We explored various combinations of the factors described above in an attempt to characterize the veracity of this technique for application to expanding bilingual lexicons (as a first step toward tri- and n-lingual

[7] Sometimes rigid designators also occur as translations, but those would not be usable to the Word-Name view.

lexicons). Except where noted, the procedure is a 100-fold cross-validation experiment. In other words, in order to test the effectiveness at adding a new word correctly to the lexicon, based solely on the S and L matrices, we ablated roughly 1% of the rows and columns in the L matrix and attempted to recover the information encoded in them. Specifically, 1% of the tuple list was selected for ablation. Any entry in L for each of the iotas present in those tuples was ablated, as if no knowledge was available regarding the translation of this iota (this with a careful eye such that the procedure would provide a fair estimate despite rampant polysemy, as noted in Section 5.2.5).

Each iota is represented by a row in S. In our running example, this would be the frequency of co-occurrence of a French iota $i_{\mathbb{F}}$ with all other French iotas in $\mathrm{m}_{\mathbb{F}}$ (each represented by a vector, $v_{\mathbb{F}}$). Similarly, we have $i_{\mathbb{E}}$, an English iota represented by the frequency of its co-occurrence with all other English iotas in $\mathrm{m}_{\mathbb{E}}$. In order to compare $i_{\mathbb{F}}$ to the possible translations in $\mathrm{m}_{\mathbb{E}}$ (such as $i_{\mathbb{E}}$), the vectors must represent similar information (i.e., both represent English co-occurrence information, rather than one English and one French). Thus, we use (the unablated portions of) $\mathrm{L}_{\mathbb{EF}}$ to transform $i_{\mathbb{F}}$ to represent the English iotas equivalent to the French iotas i occurs with ($i_{\mathbb{F}}^{\mathbb{E}}$), further described in Section 5.4.1. Specifically,

$$i_{\mathbb{F}}^{\mathbb{E}} = i_{\mathbb{F}} \mathrm{L}_{\mathbb{EF}}.$$

For each i in our fold, we find

$$i_{\mathbb{E}}^{*} = \min_{\forall j \in \mathrm{m}_{\mathbb{E}}} (\delta(i_{\mathbb{F}}^{\mathbb{E}}, j_{\mathbb{E}})),$$

where δ is some distance measure, discussed in more depth in Section 5.3.1. Thus, $i_{\mathbb{E}}^{*}$ is the closest (according to some δ) English iota to the French iota $i_{\mathbb{F}}$. We compute the δs for all iotas in $\mathrm{N}_{\mathbb{E}}$. This simulates the condition where an unknown French iota can be translated to any known English iota, regardless of whether that English iota is presently in our lexicon or not (in the face of rampant polysemy, this is the use-case we envision).

5.3.1 Distance Measures

There are a variety of ways to measure the distance or dissimilarity of one iota to another, where each iota i is defined as a vector, e.g., $i' = [i_1, i_2, \ldots, i_m]$, for m dimensions in a particular view. In general, for iotas i and j, a dissimilarity measure δ for $\delta(i, j) = \delta_{ij}$ is symmetric ($\delta(i, j) = \delta(j, i)$), nonnegative ($\delta(i, j) \geq 0$), and hollow ($\delta(i, i) = 0$). The interpretation is that iotas (i, j) are more dissimilar than (i, k) if and only if $\delta(i, j) > \delta(i, s)$. Since it is not obvious which distance measure is the most appropriate one to use for each view, in our experiments we compute results from five different distance measures, $\delta_{\cos}$, $\delta_{\frac{1}{2}}$, δ_1, δ_2, and δ_∞, with the following definitions:

$$\delta(i, j)_{\cos} = 1 - \frac{< i, j >}{||i||\ ||j||}$$

$$\delta(i, j)_{\frac{1}{2}} = ||i - j||_{\frac{1}{2}} = (\sum_{k=1}^{m} |i_k - j_k|^{\frac{1}{2}})^2$$

$$\delta(i, j)_1 = ||i - j||_1 = \sum_{k=1}^{m} |i_k - j_k|$$

$$\delta(i, j)_2 = ||i - j||_2 = (\sum_{k=1}^{m} |i_k - j_k|^2)^{1/2}$$

$$\delta(i, j)_\infty = ||i - j||_\infty = \max_{k=1,\ldots,m} (|i_k - j_k|).$$

5.3.2 Fusion

We hypothesize that using various views of the data in concert would allow for better (more powerful) inference, and thus a better lexicon induction technique. One method of fusion (end-hoc fusion) was examined during the current investigation, but immediate follow on work will expand this to other fusion approaches. Ranked lists ($r \in R$) of candidate translations for a given iota can be obtained from any one of the distance measures described in Section 5.3.1, so r_∞ is the ranked list based on the ∞ norm and $r_\infty(i)$ is the rank of iota i in r_∞. Unless otherwise specified, $R = \{r_{\frac{1}{2}}, r_1, r_2, r_\infty, r_{\cos}\}$. One method of fusion (end-hoc) is to combine many r into a single fused ranked list (r^*), representing some combination of the rankings, e.g., :

$$s(i) = \sum_{r \in R} r(i).$$

The ranking in r^* is determined by ordering the i in ascending order according to $s(i)$.

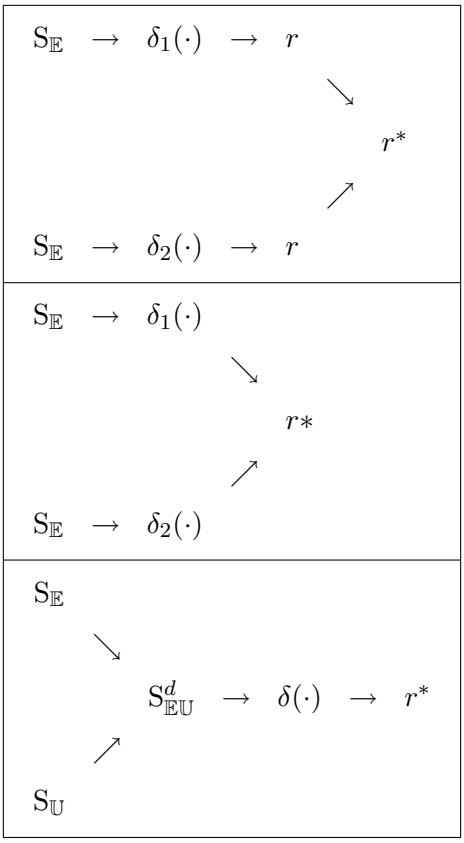


Figure 5.2: Different approaches to fusion of disparate data: (top) Measure, rank, then fuse, (middle) Measure, fuse, then rank (also referred to as end-hoc fusion), and (bottom) embedding. Raw data is represented by S, measurements on the data are represented by $\delta(\cdot)$, r are ranked lists, and r^* is the final ranking. "Measure, rank, then fuse" in this case is equivalent to reranking: combining the ranked lists. "Measure, fuse, then rank" operates on the distance measures directly, in concert, to derive a ranked list. "Embedding" is further described in Section 5.3.3 and fuses at the level of the raw data.

5.3.3 Embedding

It is noted that the iotas represented as S-Matrices, Section 5.2.4, are high-dimensional objects. Further, iotas in different languages and across different views live in different high-dimensional spaces. Thus, embedding the iotas in low-dimensional, common spaces allows for intuitive and interpretable representations amenable to statistical comparison and intuitive generalization and application.

To be more precise, if the original iota space is R^N, then to embed an $N \times N$ dissimilarity matrix $\Delta = [\delta_{ij}]$ in R^d means finding a set of points $x_1, \ldots, x_N \in R^d$ such that $||x_i - x_j|| \approx \delta_{ij}$. Generally speaking, we seek to place points into an embedded space such that the distance between a pair of points in the embedded (lower dimensional) space has fidelity to the distance between those points in non-embedded (higher dimensional) space (i.e., two iotas with small $\delta_{\cos}$ will be close together and two iotas with large $\delta_{\cos}$ are further apart). In general, this embedding technique is known as Multidimensional Scaling, e.g., (Borg and Groenen, 1997).

While some work has been done on combining dissimilarity representations in an embedding space, e.g., (Ma et al., 2008), we found the need to incorporate information from the views of the data, i.e., the $\mathrm{S}^i_{\mathbb{L}}$ as well as the link matrices L. Thus, if translating, for example, French into English, then we first form the transformed French-to-English view as $\mathrm{S}_{\mathbb{FE}} = \mathrm{S}^i_{\mathbb{F}}\mathrm{L}_{\mathbb{EF}}$. Now the vectors in $\mathrm{S}_{\mathbb{E}}$ and $\mathrm{S}_{\mathbb{FE}}$ are in the same space.

We can then construct the $M^{(2)}$ matrix of squared Euclidean distances between all iota vectors in

$$M = \begin{bmatrix} \mathrm{S}_{\mathbb{E}} \\ \mathrm{S}_{\mathbb{FE}} \end{bmatrix}.$$

To perform Classical Multidimensional Scaling (Borg and Groenen, 1997), if $e = (1, \ldots, 1)^t \in R^N$, I is the $N \times N$ identity matrix, and $P = I - ee^t/N$ is a centering projection matrix, then we form the matrix of scalar products B ((Trosset and Priebe, 2008)) as

$$B = -PM^{(2)}P/2.$$

From the singular value decomposition $B = U\Sigma^2U^t$, where $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_N)$ and $\sigma_1 \geq \cdots \geq \sigma_N > 0$ are the singular values of B, then the best rank-d approximation of the original N objects, in the least-squares sense, is given by the d-dimensional vectors $U_d\Sigma_d$.

We can then assess the embedding performance by measuring the distances between corresponding iotas in the embedding space.

5.4 Methods

We seek to create and expand a bilingual lexicon, either between English / French[8] or between English / Urdu, since adequate monolingual data is readily available for these pairs. Thus, $\mathbb{L} = \{\mathbb{E}, \mathbb{F}\}$ or $\mathbb{L} = \{\mathbb{E}, \mathbb{U}\}$[9]. All results reported use hand-curated dictionaries as truth. We did obtain alignment-induced dictionaries for comparison, though we have not conducted an unbiased experiment between these two "truth" sets yet.

[8] At least one of us speaks a modicum of French. This allows for easy error checking and the disambiguation of problems due to data-set-encoding (as was the case initially for Urdu) and problems due to the method itself. Using French also sets the stage to do multi-lingual lexicon induction, using the Europarl corpus, where all the European languages are represented.

[9] Some related preprocessing steps can be found in a (yet to be written) paper regarding the exploitation of various metadata for this task.

5.4.1 Test Procedure

To evaluate the efficacy of discovering the translation of an iota from its context alone, we simulate this technique using iotas for which we have a known translation (derived from entries in our hand-curated dictionary). For illustrative and notational purposes, the example we present is finding the English translation for a French word. For French iota $i_{\mathbb{F}}$, we seek its English translation ($i_{\mathbb{E}}$). Co-occurrence information for a given view (e.g., 5-window) is represented as a vector ($v_{\mathbb{E}}$ or $v_{\mathbb{F}}$); this is the row corresponding to i of the appropriate S matrix, so $v_{\mathbb{F}}$ is the row corresponding to $i_{\mathbb{F}}$ in $\mathrm{S}_{\mathbb{F}}$.

The L matrices, in this case $\mathrm{L}_{\mathbb{EF}}$, capture the probability of translation of iotas in the S-matrices that they correspond to (in this case $\mathrm{S}_{\mathbb{E}}$ and $\mathrm{S}_{\mathbb{F}}$). Moreover, the L-matrices are engineered such that:

$$v_{\mathbb{F}}^{\mathbb{E}} = \mathrm{L}_{\mathbb{EF}} \times v_{\mathbb{F}},$$

where $v_{\mathbb{F}}^{\mathbb{E}}$ is the English context of the French iota. Said another way, it represents the translation into English of each of the French iotas the target French iota co-occurs with. The best performance is expected where the S-matrices are derived from parallel corpora (a direct translation, thus having precisely the same context in both languages). In such a scenario, $v_{\mathbb{F}}^{\mathbb{E}} = v_{\mathbb{E}}$, since the context is as equivalent as possible (deviations supposedly due to language-specific changes, e.g., polysemy).

We use this procedure to simulate growing an existing lexicon (as is suggested in Section 5.3). To examine the performance of this technique, we collect the distance measures for each French iota to (1) the English iota corresponding to a correct translation and (2) to a random English iota. This is effectively a Monte Carlo simulation that yields a pair of distributions, one from which we expect a correct pairing to be drawn from and one from which we expect an incorrect pairing to be drawn from. The Null hypothesis, H_0, is that we have an incorrect pairing, and the alternate hypothesis, H_1, is that we have a correct pairing. Thus we can decide upon the portion of incorrect translations admitted (α, type 1 error) to determine a critical value at which to accept or reject a given iota pairing to our lexicon. Thus, setting $\alpha = 0.05$ would establish a critical value that would admit 5 percent of incorrect translations to the lexicon. The power of the technique ($1 - \beta$ at level α) is the probability of detecting a correct translation. For brevity, and since our explorations will only give empirically-derived estimates of power, we use $\hat{\beta}$ to refer to the power at level α.

5.4.2 Metrics of Performance

When facing rampant polysemy, judging solely on rank of a given translation of an iota is problematic. Thus, performance is reported either as the area under a ROC curve and/or as an estimate of statistical power via comparison of correctly- and randomly-paired iotas. The null hypothesis is that a given pair of iotas are not a translation of one another, while the alternative hypothesis states that they are. In such a scenario, a threshold is established below which (since we are dealing in distance metrics) the iotas are assumed to be a translation for one another and above which they are not. The ROCs plot the precision (portion of rejected hypothesis that are correct) on the ordinal against the recall (portion of the misses or falsely accepted null hypotheses). Further, the plots of the distances of correctly- and randomly-paired iotas show the spread of these distance metrics. In these paired histograms, the critical value at a few choices for the level of α are shown along with an estimate of the power ($\hat{\beta}$) corresponding to that critical value.

5.5 Results

The results presented here are from a single co-occurrence metric (a 5-window) and a single language pair (either English/Urdu or English/French). Figure 5.3 shows far superior performance for $\delta_{\cos}$ when compared to the other distance metrics investigated. Thus, the remaining figures and investigations were based on

using $\delta_{\cos}$ alone. It should be noted, however, that the other distance metrics still provide some amount of information, and thus may still be useful when fused with $\delta_{\cos}$, as preliminary results (unreported here) suggest.

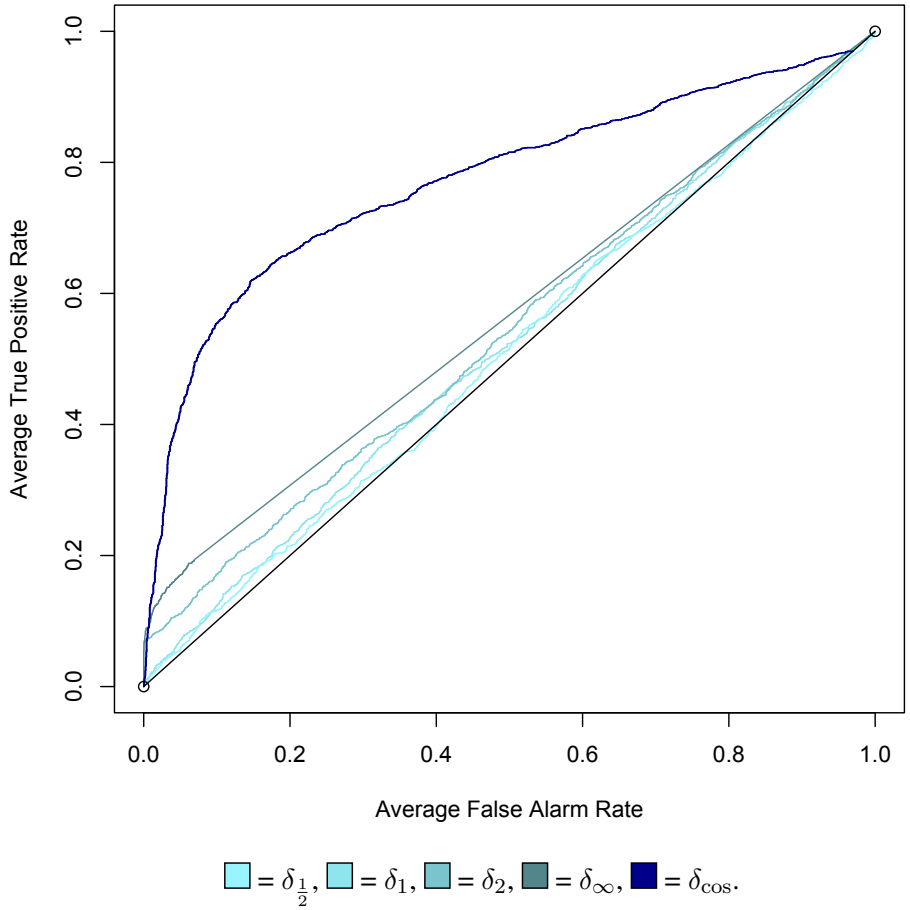


Figure 5.3: The ROC curve for the English / French Monolingual Comparable Corpora, using various distance measures to compute similarity. Clearly, $\delta_{\cos}$ dominates the rest.

Though a number of normalization options for the S- and L-matrices were possible, as outlined in Sections 5.2.4 and 5.2.5, we found that performance was roughly equivalent between the few we initially investigated (a pair of which is presented here). This issue of normalization deserves further investigation, but preliminary results indicate that normalizing L such that rows sum to 1 and not performing such normalization, so the resulting L is binary, with $\mathrm{L}(i, j) = 1$ if i is a translation of j and 0 otherwise, are statistically equivalent. This lack of statistically significant difference can be seen in Figure 5.4.

An upper bound for power was established by testing the technique on the Europarl parallel corpus (for English/French). A histogram showing the paired and unpaired distributions can be found in Figure 5.5. Importantly, this shows that the information captured by the single co-occurrence metric and single distance measure would not be sufficient to correctly pair all of the iotas, even when presented with a parallel corpus (where we are certain the contexts are identical across languages). Thus, when using comparable corpora, as our use-case dictates, we cannot hope to achieve perfect pairings. This can be seen in Figure 5.6. Comparing

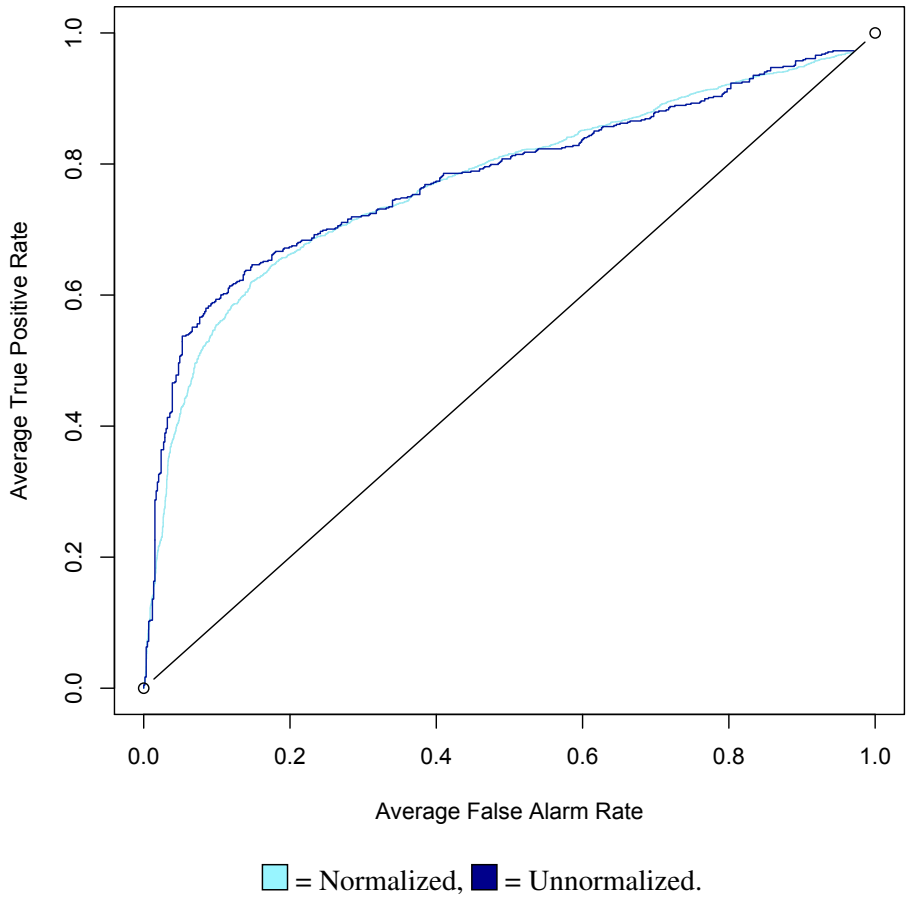


Figure 5.4: The ROC curve for English / French Monolingual Comparable Corpora, using $\delta_{\cos}$. For both curves, $S_{\mathbb{E}}$ and $S_{\mathbb{F}}$ were normalized. For the light curve, $L_{\mathbb{EF}}$ was not normalized, and for the dark curve $L_{\mathbb{EF}}$ was normalized. The difference is not statistically significant.

the estimated power of the tests at $\alpha = 0.05$, we see that $\hat{\beta} = 0.70$ for the English/French parallel corpus (Figure 5.5), while for the English/French monolingual corpora it is only 0.45 (Figure 5.6).

The equivalent histogram for the English/Urdu monolingual corpora is shown in Figure 5.7, with a power of only 0.29 (at $\alpha = 0.05$). At this level, the performance is significantly poorer for English/Urdu than English/French, but quite the opposite is usually the case (as shown by the ROC curves in Figure 5.8).

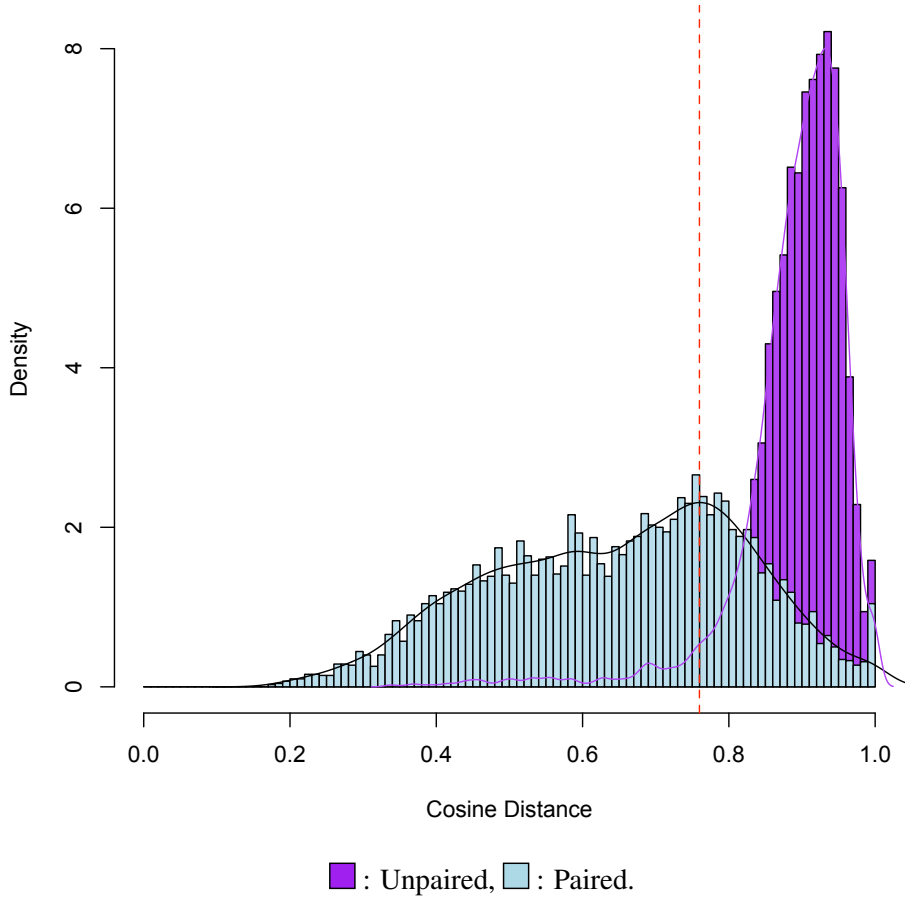


Figure 5.5: Histogram of paired/unpaired distances for the Europarl (parallel) English/French corpus. An estimated density overlays the histogram, purple for unpaired and black for paired. The vertical dotted line indicates $\alpha = 0.05$, $\hat{\beta} = 0.70$. At $\alpha = 0.01, \hat{\beta} = 0.26$ and at $\alpha = 0.10, \hat{\beta} = 0.82$. This serves as an upper bound on performance for a single distance measure ($\delta_{\cos}$) and a single language pair.

5.5.1 Embedding

We performed a limited test of embedding a slice into a lower-dimensional space in order to assess correct word pair detection in a reduced dimensionality space. We also wanted to assess possible benefits of reducing the effects of polysemy.

From a list of most-frequent French iotas, the most-frequent French iota was chosen and the English iota with the highest probability of being the translation of that French iota, based on the values stored in the link matrix (L), was induced as the translation of that French iota with probability equal to 1. The remaining

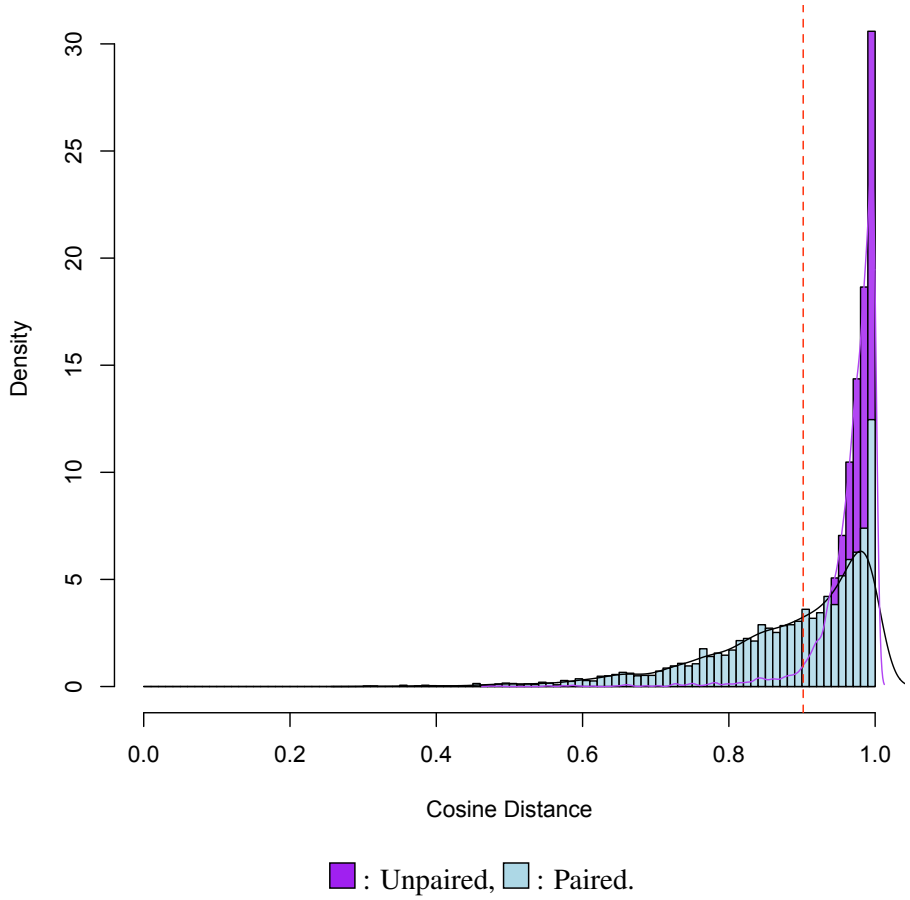


Figure 5.6: Histogram of paired/unpaired distances for ESF Monolingual Comparable Corpora. An estimate of the density also overlays each histogram. The vertical dotted line indicates $\alpha = 0.05$ and $\hat{\beta} = 0.45$. At $\alpha = 0.01$, $\hat{\beta} = 0.16$ and at $\alpha = 0.10$, $\hat{\beta} = 0.56$.

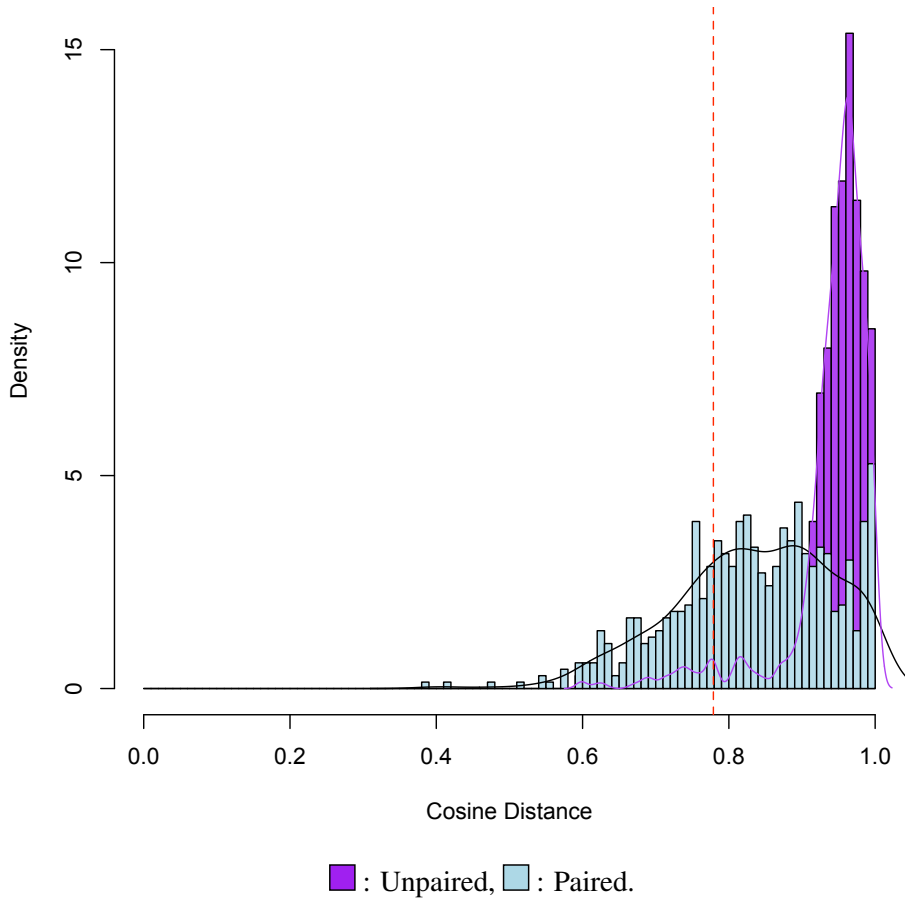


Figure 5.7: Histogram of paired/unpaired distances for English/Urdu Monolingual Comparable Corpora. The density also overlays the histogram, purple for unpaired and black for paired. The vertical dotted line indicates $\alpha = 0.05$, $\hat{\beta} = 0.29$. At $\alpha = 0.01$, $\hat{\beta} = 0.10$ and at $\alpha = 0.07$, $\hat{\beta} = 0.49$.

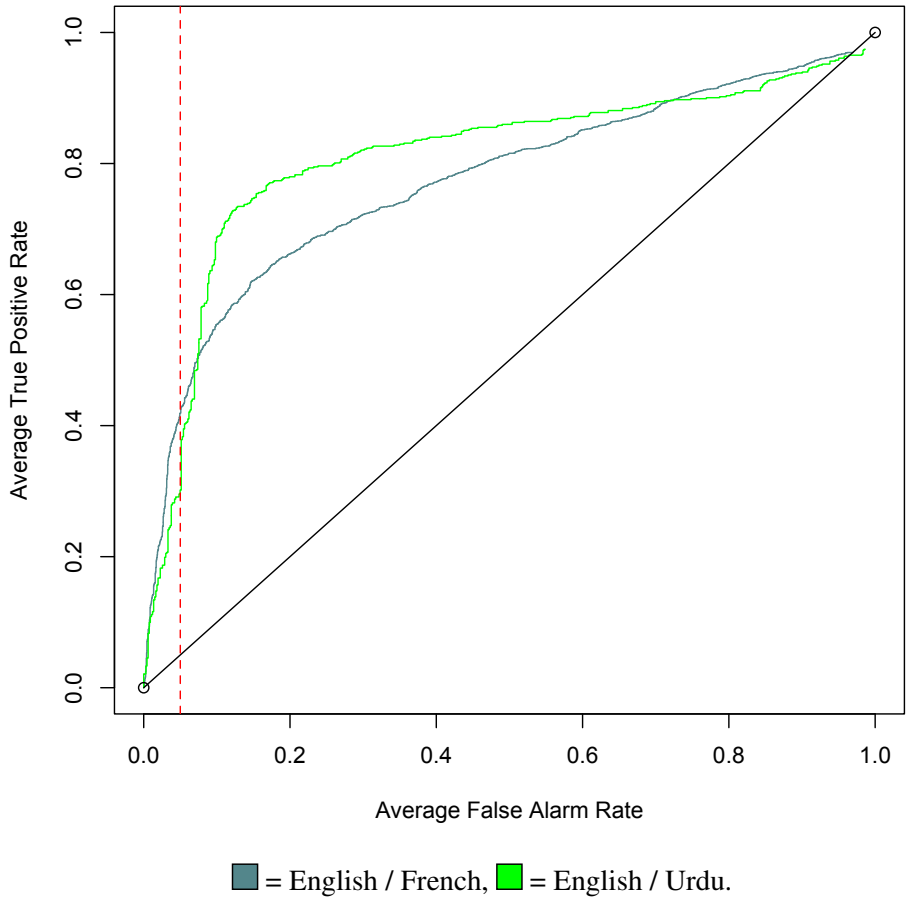


Figure 5.8: ROC curves for Monolingual Comparable Corpora, English / Urdu and English / French Slices, using $\delta_{\cos}$ as the distance measure, $\alpha = 0.05$, $\hat{\beta}(\mathbb{EF}) = 0.42$ and $\hat{\beta}(\mathbb{EU}) = 0.30$. At $\alpha = 0.01$, $\hat{\beta}(\mathbb{EF}) = 0.13$ and $\hat{\beta}(\mathbb{EU}) = 0.10$ and at $\alpha = 0.10$, $\hat{\beta}(\mathbb{EF}) = 0.55$ and $\hat{\beta}(\mathbb{EU}) = 0.68$.

French iotas in the lexicon were similarly selected in this manner. In this way, there is an induced many-to-one translation map, in that a given French iota can have only one English translation, but multiple French iotas can translate to the same English iota. This is actually a simplification of the typical many-to-many relationships between iotas in real life.

In this many-to-one English/French scenario, using the monolingual corpus, 600 duples were thus obtained. English and French word co-occurrence slices were thus created for these 600 words. As discussed in Section 5.3.3, these slices were reduced in dimensionality by embedding to 50% the original vector dimensions. Figure 5.9 shows the performance in the embedding space, and with a $\hat{\beta} = 0.865$, the result suggests embedding shows promise as a means of receiving performance benefits from performing in a reduced-dimensionality space while still receiving reasonable performance. It is important to note that these results are not directly comparable to the other results, since the embedding was done without any cross-fold validation (ablation of translations probabilities).

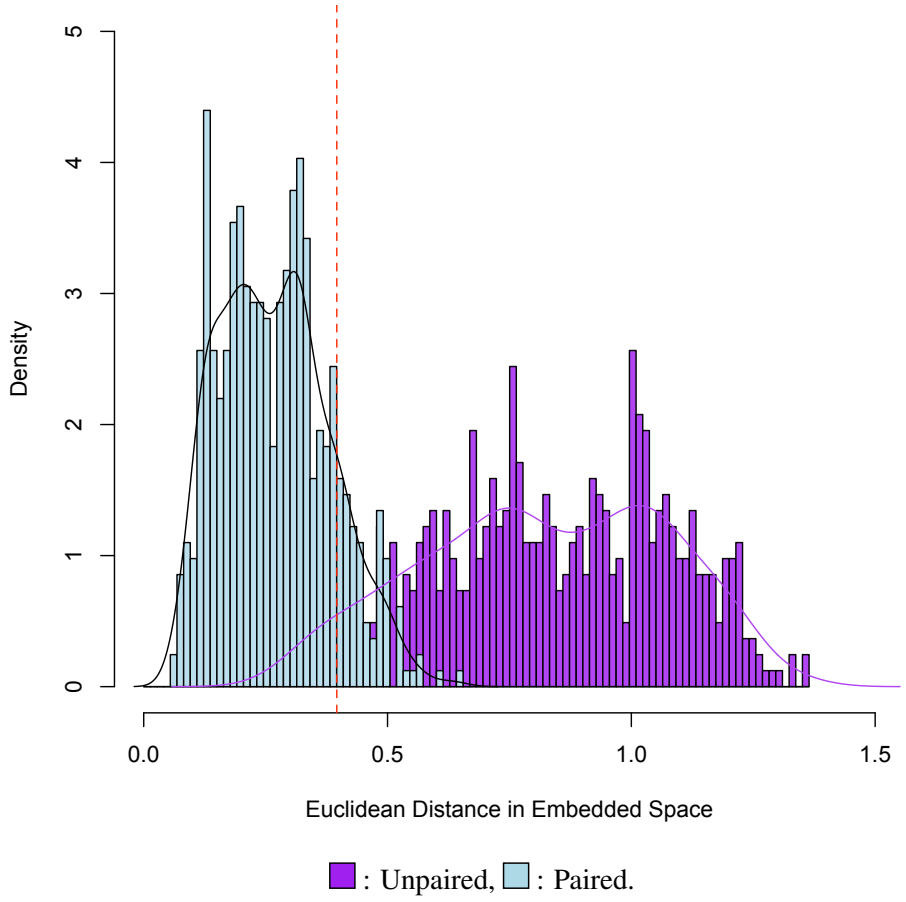


Figure 5.9: Histogram of paired/unpaired distances in an embedding space (dimensionality reduction = 50%) for the Monolingual Comparable Corpora, English/French corpus. The link matrix had an induced many-to-one polysemy. An estimated density overlays the histogram, purple for unpaired and black for paired. The vertical dotted line indicates $\alpha = 0.05$, $\hat{\beta} = 0.865$.

L was next made more strict by inducing a one-to-one relationship between the English and French iotas. That is, each French iota and only one English iota translation, induced from the most likely English

translation, and multiple French iotas could not map to the same English iota (if a duplicate English iota is found in the process of creating the lexicon, then that French iota is removed and the next French iota in the list of candidate French words is used). Similarly to the many-to-one experiment, 600 duples were selected.

Using a similar embedding process, Figure 5.10 shows the histogram between paired and unpaired words in the embedded space for the one-to-one (induced) distances. Interestingly, but perhaps to be expected, with monosemy, word translations become easier to compute, even in an embedding space. Performance degrades, as seen in the many-to-one case, with increasing polysemy.

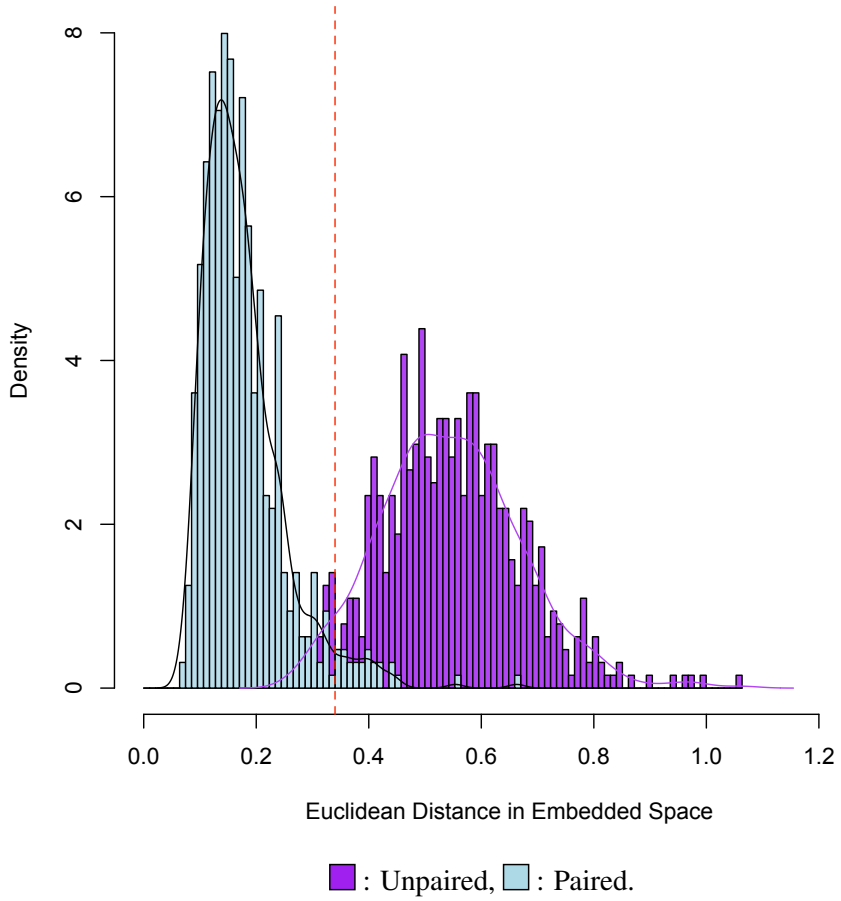


Figure 5.10: Histogram of paired/unpaired distances in an embedding space (dimensionality reduction = 50%) for the Monolingual Comparable Corpora, English/French corpus. The link matrix had an induced one-to-one polysemy. An estimated density overlays the histogram, purple for unpaired and black for paired. The vertical dotted line indicates $\alpha = 0.05$, $\hat{\beta} = 0.96$.

5.6 Discussion

The results indicate that this technique is viable, though using only a single language pair, a single co-occurrence metric, and a single distance measure does not provide sufficient information to operationalize

the technique[10].

The results, presented in Section 5.5, indicate that some information is being adequately conveyed by co-occurence and captured by this technique. Preliminary experiments expanding upon the breadth and depth of information used shows improvements. Thus, using more than one co-occurece metric, more than one distance measure, and/or more than one language pair we expect to improve performance. The framework developed this summer and presented in Section 5.1.2 is versatile and completely agnostic to the type of data used to populate the S-matrices, so the effort to expand along these lines lies almost entirely in the creation of the data views. On a related note, the theoretic framework and associated machinery has been engineered such that it can scale gracefully with the number of languages. Though we talk and experiment with up to three languages here, an arbitrary number of languages can be used.

5.6.1 Future Work

The short time frame did not allow many ideas to come to fruition, though the proof-of-concept work described herein begs for follow-on work. Here we outline ideas for the near and far term alike.

Multiple Languages

Preliminary results indicate that using multiple languages can improve the inference. Intuition provides a few reasons why this may be true: non-symmetric polysemy disambiguating word senses, non-systematic noise in each language mitigated by the others, etc.

Some questions worth pursuit are:

- Does adding another language provide for significantly better inference?
- Which of the fusion methods (Figure 5.2) provides the most power?
- How much does the relatedness between language pairs change these relationships (e.g. French / Spanish and French / Urdu are not related in the same way)?

Zymurgy

The inclusion of a different kind of language or ontology, such as Zymurgy, would allow a semantic structure to influence the structures. Conversely, Semantic Tunneling may be useful to improve the coverage of Zymurgy both within a language (that has not received as much attention as English) or to new languages for which a Zymurgy lexicon has not been established.

String Edit Distance

String edit distance is a measure of how far apart two strings are, based on the number of operations needed to transform one string into another. We expect that, especially in the multilingual case, that computing views of the data based on string edit distance will be helpful, since we expect words from within a family of languages, such as Hindi, Nepali, and Punjabi, to have smaller string edit distances than words from another language family, such as Czech, Polish, and Russian (Schafer and Yarowsky, 2002b). We conjecture that this is especially true for languages like Urdu, which is a conglomeration of a number of disparate languages and likewise draws iotas from each of them. Being able to tell that a given iota is like a known Arabic iota and unlike any Hindi iotas would be evidence in favor of using Arabic as a bridge to English,

[10]We present results here obtained over the 8 weeks of SCALE, though there was not sufficient time to fully explore the fusion and embedding aspects of this work, where initial results were most promising.

rather than Hindi alone or some fusion of Hindi and Arabic. In a sense, this could also be used to condition (in the statistical sense) the way in which the other slices are interpreted, fused, and used.

Matrix Manipulation for Unsupervised Dictionary Creation

An investigation was made into how well words in one language translate into another language using only matrix manipulation.

Let us assume that word co-occurrence between languages remains largely unchanged. We assume this because both human psychology and the physical world constrain the set of possible actions. This in turn shapes language usage and word co-occurrence probabilities. Take cereal for example. Humans generally pour cereal into a bowl add some milk and eat it with a spoon. They also buy cereal at a supermarket. They generally do not throw cereal, or use it as a pillow. Similarly they generally do not talk about doing these things. The use of cereal in both the real world and in language is constrained by its intended function.

We assert that the source and target languages are the same ideas that are just expressed with different symbols in a different order. This would mean that the symbols are just permutations of each other.

We can express each language as a co-occurrence matrix by using a sliding window to create a word by word co-occurrence matrix. This matrix forms the basis with which documents are represented in that language. We assert that these two basis representations are two ways to represent the same idea or message. In this vein we can translate documents, sentences, and words by representing them as a matrix and then applying the correct permutation. However, translation is more complicated than just multiplying the source matrix S times the permutation matrix B to get the target language matrix T because we must permute both the rows and the columns of S. Take a simple 3 by 3 matrix. In this matrix we have three words arranged in order along the rows and columns. Let us pretend that the first source word translates to the third target word, that the second source word translates to the first target word, and that the third source word translates to the second target word. Let us examine multiplying it with a simple permutation matrix:

$$\begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} B & C & A \\ E & F & D \\ H & I & G \end{pmatrix}$$

Reading along the columns you encounter the correct values but in the wrong order. The correct mathematical technique to use in this situation is the change of basis. Now let us apply the transformation on both sides like $B_t S B = T$ to complete our work.

$$\begin{pmatrix} B & C & A \\ E & F & D \\ H & I & G \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} E & F & D \\ H & I & G \\ B & C & A \end{pmatrix}$$

As you can see the first row has been moved down to the third row, and its elements have been reordered according to the word permutation given. This is the correct answer because the values on the diagonal is the count that that word occurred with itself. These values have been reordered to match the translation provided above. Using the permutation on both sides allows us to correctly permute both the rows and the columns to produce a target matrix where both the row index and the column index mean the same thing.

You might object that we used the transpose of B instead of its inverse, but this is mathematically justified. Since the source and target matrices are symmetric, which means that their eigenvector matrices

V_s and V_t are orthonormal, $V_s^t = V_s^{-1}$ and $V_t^t = V_t^{-1}$.Therefore, we can conclude that $B^{-1} = B_t$:

$$B = V_t V_s^{-1} = V_t V_s^t$$
$$B^{-1} = V_s V_t^{-1} = V_s V_t^t$$
$$B^t = (V_t V_s^t)^t = V_s V_t^t$$

So we can save time by computing the transpose of B instead of its inverse.

This kind of change of basis can be performed to represent the documents from one language in the language of the other set. We assert that these two basis representations are two ways to represent the same idea or message. Under this framework translation is just the representation of the same idea in a different distribution. To translate a document correctly we need only represent the ideas in the target distribution and then arrange the translated words into the correct order. To translate between the vocabulary in the source language S, and the target language T, we need only calculate and apply a change of basis B. So by applying B to S as follows we can represent a source language matrix in terms of the target language since: $BSB_{-1} = T$

Our derivation follows. Given that matrices S and T are similar, then their eigenvalues should be the same by definition. So their diagonal matrices should be the same. So: $D_s = D_t$. Let us call the eigenvector matrix of matrix S V_s, and the eigenvector matrix of T V_t. Let us multiply both side of the equation by some useful matrices to get to a more useful form.

$D_s = D_t$

$D_s V_{t-1} = D_t V_t^{-1}$

$V_t D_s V_t^{-1} = V_t D_t V_t^{-1}$

Since $V_t D_t V_t^{-1}$ is the eigenvalue decomposition of T we can substitute the equation $T = V_t D_t V_t^{-1}$ in. So:

$V_t D_s V_t^{-1}$ = T

Now let us add in the identity matrix I in two important positions.

$V_t I D_s I V_t^{-1}$ = T

Since $V_s^{-1} V_s = I$ and since $V_s V_s^{-1} = I$ by the definition of an identity matrix:

$V_t V_s^{-1} V_s D_s V_s^{-1} V_s V_t^{-1} = T$

Use the definition of an Eigenvalue decomposition to create:

$V_t V_s^{-1} S V_s V_t^{-1}$ = T

Since $(AB)^{-1} = B^{-1} A^{-1}$ by the properties of an invertible matrix,
and since Eigenvalue matrices are linearly independent and therefore invertible we create the substitution $B = V_t V_s^{-1}$, and its inverse $B^{-1} = V_s V_t^{-1}$. Resulting in:

$BSB^{-1} = T$

This depends upon our initial assumption being correct. Are S and T really the same thing represented in different ways? We can check this assumption by comparing the eigenvalues of the two distributions. The diagonal of the D matrix contains the eigenvalues for that decomposition, the other values are zeros. A cosine of the two diagonals should tell us how well they match. Using the transcripts of the European Parliament, which are translated by hand into several languages, we attempted to verify this assumption. Our tests used Spanish as the source matrix and English as the target matrix. The cosine of their eigenvalues was .9367. A one would indicate that our matrices are similar, our score shows that our assumption is close to correct. Therefore, attempting to apply B to document translation will result in an imperfect translation. Since the eigenvalues don't perfectly align this causes a distortion.

We also used a corpus of world news in both English and Spanish collected during the same time period. These two corpora are not direct translations of each other like the European Parliament transcripts, but their eigenvalues have a better cosine of .9926. I'm not exactly sure why it came out better. It could be because this news subset was taken from a particular category of news forcing the articles to share a greater degree of similarity.

It is also possible that our assumption still holds, but that our artificial limit on the matrix dimensions excluded parts of the full matrix that would provide corresponding eigenvalues resolving any misalignments because of size limitations. This is quite reasonable, because we limited which words to represent in our matrix to words that we could check against a hand crafted bilingual dictionary of about a few thousand words. It is possible that some of the entries in the dictionary are missing, incomplete, or inaccurate. This would cause us to exclude concepts from either the source or the target basis that might exist in the other basis.

Accepting that there will be some misalignment, it should still be possible to project from one basis to the other using the matrix B. On the news dataset after projecting into the target basis we achieved an average row by row cosine similarity with the target distribution of 0.9827421971724286. On the European Parliament our projected matrix had a cosine similarity of 0.9968972846223788 with the observed target distribution. Clearly, any distortions caused by the above causes is relatively minor at the dataset level.

These distortions could be caused by a variety of factors ranging from social and cultural factors to environmental factors such as the local climate and resources. Although the exact causes are unknown we can correct for them by applying a new matrix M to adjust for these differences. Let us apply some diagonal matrix M to make $MD_s = D_t$. In this case $M = D_t D_s^{-1}$. By replacing our old starting point with $MD_s = D_t$ we can conclude that $V_t M V_s^{-1} S V_s V_t^{-1} = T$. Since all of the parts on the left and right of S are known we can project sentences in the source basis S to the target basis T.

We can use the B matrix to help us translate words. The B matrix is not a direct translation lookup table. Since word context determines meaning we need to translate words in context. To do this we create a source language matrix representing a given context by counting the co-occurrences in that context.

So far sentence projection using this technique has yielded inconclusive results. We limited our vocabulary size by a list of known word translations so that we could check our results against human judgments. This resulted in the average sentence of 20 non-unique words having only about 5 non-unique words that we knew translations for. In practice this results in many sentences being represented as having only a few unique words. Since these words are spread out in the sentence this also resulted in fewer word co-occurrences in our sliding window of 5 words. These factors make it hard to judge the success of our technique because even human observers can not determine the meaning of the sentences. Even preserving word order does not allow humans to preform well.

While this technique is effective at projecting the co-occurrence statistics of a corpus in one language into another language we are unsure how well this will work for lower data volumes such as document, paragraph, and sentence level analysis. Our tests at the sentence level are inconclusive because our baseline human performance using our evaluation dictionaries is so poor. We have not yet tested the document and paragraph level approaches.

Chapter 6

Active Learning for Statistical Machine Translation

Statistical machine translation (SMT) systems rely on large amounts of translated text during their training phases. Without sufficient amounts of training data, the performance of SMT systems can be poor. However, creating translated text as training data for SMT systems is a labor-intensive and expensive process, especially for language pairs where translators are not easy to find. An example of such a language pair is Urdu-English, the pair on which we focus our experiments.

The Linguistic Data Consortium (LDC) has created a language pack[1] containing training data for Urdu-English consisting of $\approx$ 88000 Urdu sentences translated into English. Another way of viewing the size of the dataset is $\approx$ 1.7 million Urdu words translated into English. In addition to this, the language pack contains an Urdu-English dictionary containing $\approx$ 114000 entries.

Figure 6.1 shows the learning curve for the Joshua MT system[2] for the LDC data (the dictionary is always used as training data and the number of sentences from the language pack is grown via random selection. Figure 6.2 shows the learning curve when training data size is measured in terms of the number of foreign words that have been translated. A main observation to make from these plots is that translation performance rises quickly at first but then a period of diminishing returns occurs: put simply, the curve flattens.

For other NLP tasks, active learning (AL) methods have been shown to be successful for reducing the annotation effort required to obtain given levels of performance (e.g., (Baldridge and Osborne, 2008; Bloodgood and Vijay-Shanker, 2009b)).There have been two very recent papers on applying AL to SMT (Haffari et al., 2009; Haffari and Sarkar, 2009). We build on and enhance that work to improve the effectiveness of AL for SMT. In particular, we pay attention to measuring annotation effort carefully and consider the difficulties MT systems may have with aligning new training data correctly. Motivated by these considerations, we propose an AL framework that solicits translations for only parts of sentences instead of full sentences.

Section 6.1 discusses related work; Section 6.2 explains the methods we use in detail; Section 6.3 presents experimental results; Section 6.4 concludes and discusses future work.

6.1 Related Work

Haffari et al. (2009) applies AL to statistical phrase-based MT for EuroParl data for translation between Spanish-English, German-English, and French-English. They select sentences to be added to the training data via a few different methods but their best methods basically select sentences which contain phrases

[1] LDC Catalog No.: LDC2006E110.

[2] http://www.cs.jhu.edu/ ccb/joshua/index.html

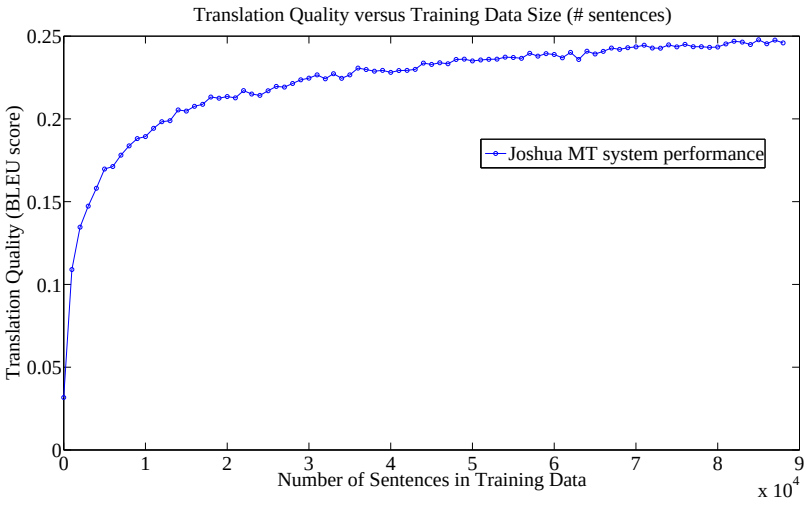


Figure 6.1: Learning curve for the entire LDC dataset. The x-axis measures the number of sentences in the training data. The y-axis measures the performance of the Joshua MT system with a 5-gram language model.

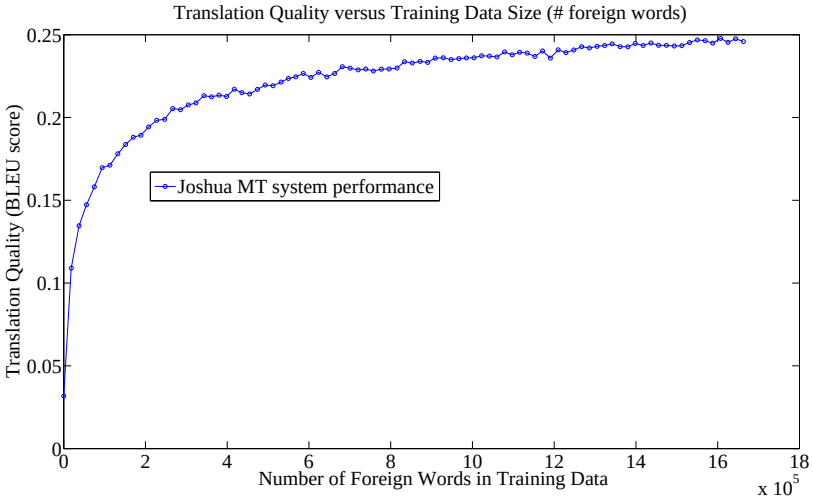


Figure 6.2: Learning curve for the entire LDC dataset. The x-axis measures the number of foreign words in the training data. The y-axis measures the performance of the Joshua MT system with a 5-gram language model.

which occur relatively frequently in the unlabeled data and relatively infrequently in the labeled data. They terminate their simulations after only 10,000 sentences in total have been annotated. In contrast, we will work on Urdu-English with multiple MT systems beyond phrase-based systems and will show results in our simulation for the entire LDC language pack ($\approx$ 88000 sentences). Haffari et al. (2009) also experiment with non-simulated AL for Bangla-English but they only add 500 sentences of training data and the differences in BLEU score observed are small. In contrast, in our non-simulated experiments we add 1000s of additional sentences and phrases of training data. Also, we gather translations for only parts of sentences and measure costs of annotation in more accurate ways than just the number of sentences that have been translated since not all sentences will have equal annotation cost (since some sentences might be long and difficult to translate versus others which are short and easy to translate). Also, we are not aware of any previous work that considers how word alignment may interact with AL methods for SMT, which we consider.

Haffari and Sarkar (2009) focuses on applying AL to the multilingual setting where one already has a parallel corpus for multiple languages and wants to add in a new language and build MT systems from each of the existing languages to the new language. This is not the setting we considered during the workshop.

Hachey et al. (2005) consider the effect of selective sampling on annotation effort for parsing tasks. Haertel et al. (2008) argue for the importance of assessing annotation costs accurately during AL and develop models for doing so for part of speech tagging tasks. In contrast to previous work, we consider how to assess annotation costs for SMT tasks.

6.2 A New Approach for Active Learning for Statistical Machine Translation

6.2.1 Annotation Effort and Word Alignment Considerations

Our goal is to develop an active learning approach that will enable us to get the highest translation quality for given levels of annotation effort. There is a great deal of literature on how to measure translation quality. We use the commonly used BLEU score and do not propose new MT evaluation metrics although we will note that BLEU score will at times understate the effectiveness of our AL approaches.

For measuring annotation effort, we consider three ways: number of sentences, number of foreign words, and number of US dollars spent. Using number of foreign words and money as cost measures are novel for AL research applied to SMT; however, we prefer them to using the number of sentences in the training data since sentences can have very different annotation costs depending on how long and difficult the sentence is to translate.

Figure 6.3 shows the learning curves for three different selection methods (random selection, shortest sentence selection, and longest sentence selection) when annotation cost is measured in terms of number of sentences. Figure 6.4 shows the learning curves for the same three selection methods when annotation cost is measured in terms of number of foreign words.

There are a few points of discussion that these figures raise. The first is that they indicate the importance of how annotation cost is measured. One would draw different conclusions about which selection methods work better depending on which annotation cost measure is used. We posit that number of foreign words is a better cost measure than number of sentences because sentences can have large variations in translation difficulty but words will have less variation. Thus, we use number of foreign words as our cost measure instead of number of sentences in subsequent experiments.

A second point is that longest sentence selection performs poorly when annotation cost is measured in terms of number of words. We hypothesize that this is because so many of the alignments will be wrong for those long sentences that less learning (per word) will take place than from shorter sentences where alignments are more likely to be correct. This remains to be verified in future work.

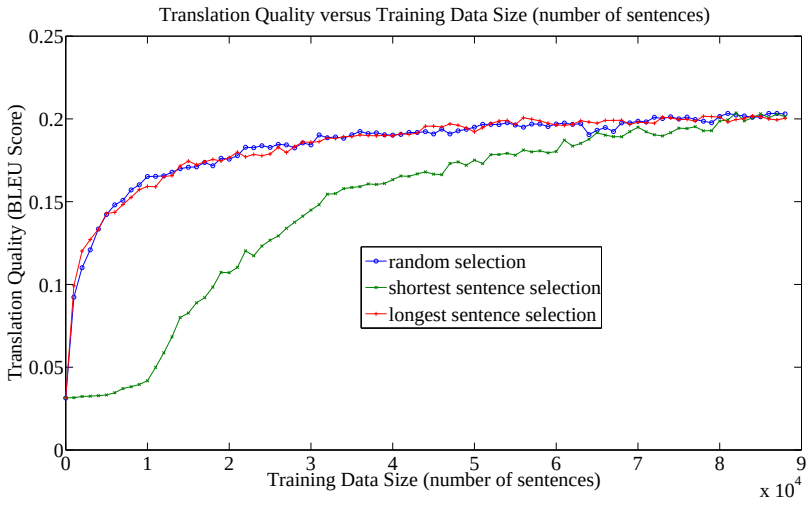


Figure 6.3: Learning curve for the entire LDC dataset for three selection methods. The x-axis measures the number of sentences in the training data. The y-axis measures the performance of the Joshua MT system with a trigram language model.

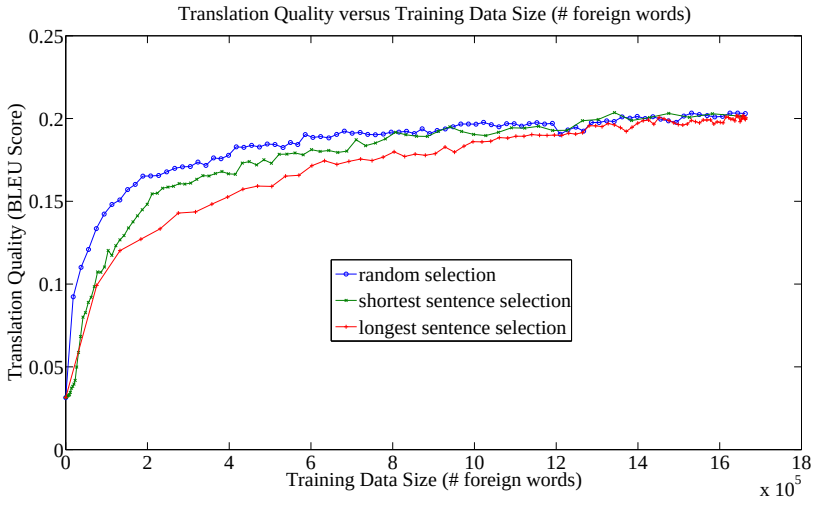


Figure 6.4: Learning curve for the entire LDC dataset for three selection methods. The x-axis measures the number of foreign words in the training data. The y-axis measures the performance of the Joshua MT system with a trigram language model.

Init:

Go through all available training data (labeled and unlabeled) and obtain frequency counts for every n-gram (n in $\{1, 2, 3, 4\}$) that occurs.
$sortedNGrams \leftarrow$ Sort n-grams by frequency in descending order.

Loop

until all n-grams in $sortedNGrams$ are covered:

1. $trigger \leftarrow$ Go down $sortedNGrams$ list and find the first n-gram that isn't covered in the so far labeled training data.
2. $selectedSentence \leftarrow$ Find a sentence that contains $trigger$.
3. Remove $selectedSentence$ from unlabeled data and add it to labeled training data.

End Loop

Figure 6.5: The n-gram sentence selection algorithm

6.2.2 N-Gram Growth Sentence Selection

Our first AL method is to select sentences for translation that contain n-grams that don't occur at all in our so-far labeled training data. We call an n-gram 'covered' if it occurs at least once in our so-far labeled data. Our algorithm has a preference for covering frequent n-grams before covering infrequent n-grams, based on the premise that the more frequent an n-gram is, the more important it is so we want to make sure we learn about it as soon as possible. The problem of automatically detecting when to stop AL is a substantial one, discussed at length in the literature (e.g., (Bloodgood and Vijay-Shanker, 2009a; Schohn and Cohn, 2000; Vlachos, 2008)). In our simulation, we stop AL once all n-grams have been covered for n in $\{1, 2, 3, 4\}$. In our experiments with human annotators, we stopped gathering translations when the workshop ended. Our algorithm for n-gram growth sentence selection is depicted in Figure 6.5.

6.2.3 Highlighted N-Gram Selection

Our next AL method solicits translations only for trigger n-grams and not for entire sentences. We provide sentential context, highlight the trigger, and ask for a translation of just the highlighted trigger. A screenshot of our interface is depicted in Figure 6.6. We ask for translations for triggers in the same order they're encountered by the algorithm in Figure 6.5.

Our motivations for soliciting translations for only parts of sentences are twofold, corresponding to two possible cases. Case one is that a translation model learned from the so far labeled training data will be able to translate most of the non-trigger words in the sentence correctly and thus by asking a human to translate only the trigger words, we avoid wasting human translation effort. Case two is that a translation model learned from the so far labeled training data will (in addition to not being able to translate the trigger words correctly) also not be able to translate most of the non-trigger words correctly. One might think then that this would be a great sentence to have translated because the machine can potentially learn a lot from the

Figure 6.6: Screenshot of the interface we used for soliciting translations for triggers.

translation. Indeed, much of AL research in the past has been based on querying examples for which the machine is most uncertain. For the case of SMT, however, we think too much uncertainty could in a sense overwhelm the machine and it's better to provide new training data in a more gradual manner. The reason is that a sentence with large numbers of unseen words is likely to not get word-aligned correctly during training and then learning will be hampered. By asking for a translation of only the trigger words, we expect to be able to circumvent this problem.

6.3 Experiments and Analysis

6.3.1 Simulated Experiments

We simulated AL on the LDC language pack data by pretending we didn't have translations and then revealing the translations as the sentences were selected for annotation. In all of our experiments that we report here, we used the Urdu-English dictionary at each round of training and grew the training data via adding sentence translation pairs. This in some sense makes it harder for us to show AL providing advantages relative to random selection since the dictionary resource will provide a common higher base for all the selection methods, including random. However, it seems the more realistic course to take since it would be artificial to ignore dictionary resources when they exist.

Figure 6.7 shows the performance of n-gram growth sentence selection versus random sentence selection on the LDC data with training data size measured in terms of sentences. We can see that the n-gram growth selection algorithm is superior. Figure 6.8 shows the performance of n-gram growth sentence selection versus random sentence selection on the LDC data with training data size measured in terms of foreign words. Although this makes the difference between n-gram growth selection and random selection look smaller, we think it is the more appropriate plot to examine. The n-gram growth selection tends to pick

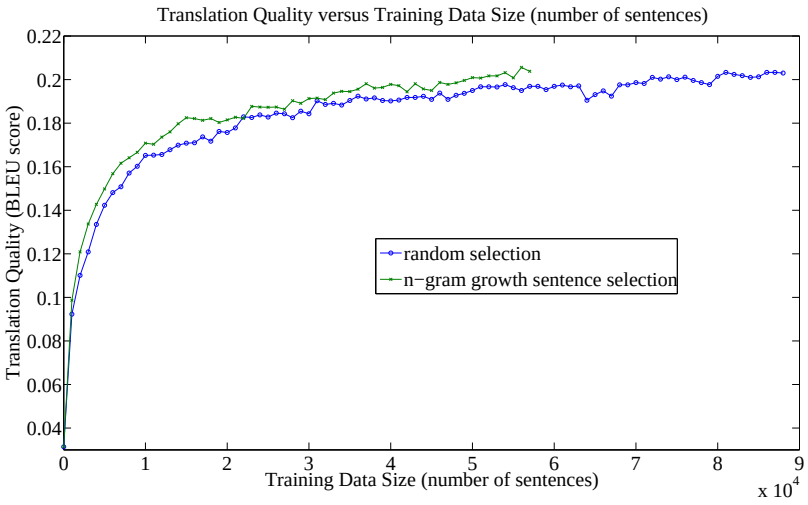


Figure 6.7: Performance of n-gram sentence selection versus random sentence selection with training data size measured by number of sentences. The y-axis measures the performance of the Joshua MT system with a trigram language model.

longer sentences than random selection does and therefore those sentences might require more translation effort. Nonetheless, even with training data size measured in terms of the number of foreign words, the n-gram sentence selection method outperforms random selection. Furthermore, the stopping technique appears to work well. Figure 6.9 shows a close-up of the plot in the area around where the n-gram selection method stops requesting more annotations. At the stopping point, only 1,161,750 words have been translated and a bleu score of 20.02 is reached with n-gram selection. Random selection doesn't achieve a bleu score of 20.02 until 1,625,135 words have been translated. The entire dataset has 1,662,638 words and using all of the data gets a bleu score of 20.20. The n-gram selection method stops with only 70% of the words translated but gets 99% of the translation quality.

6.3.2 Non-simulated Experiments

Encouraged by the simulation successes, we next set out to see whether we could achieve translation quality improvements by gathering additional translations to add to the training data of the entire LDC language pack. In particular, we wanted to see if we could achieve translation improvements with a relatively small annotation expense. Note that at the outset this is an ambitious endeavor (recall the flattening of the curve in Figure 6.2).

We used the Amazon Mechanical Turk (AMT) web service to gather translations in a cost-effective manner. We crawled BBC articles that were in Urdu and applied the highlighted n-gram selection algorithm from Section 6.2.3 to determine what to post on AMT for workers to translate[3]. We gathered an additional 5806 n-gram translations at a cost of one cent per translation, giving us a total cost of $58.06. In terms of words, we acquired translations for 16,981 Urdu words. Note that this is a relatively tiny amount, $\approx 1\%$ of the LDC data. Figure 6.10 shows the performance when we add this training data to the LDC corpus. Observe that the rectangle around the last 450,000 words of the LDC data is wide and short but the rectangle around the newly added translations is narrow and tall. This gives promise for the approach being able to address the challenge posed by the flattening of the curve that was discussed at the beginning of this paper.

[3]For practical reasons we restricted ourselves to not considering sentences that were longer than 60 Urdu words, however.

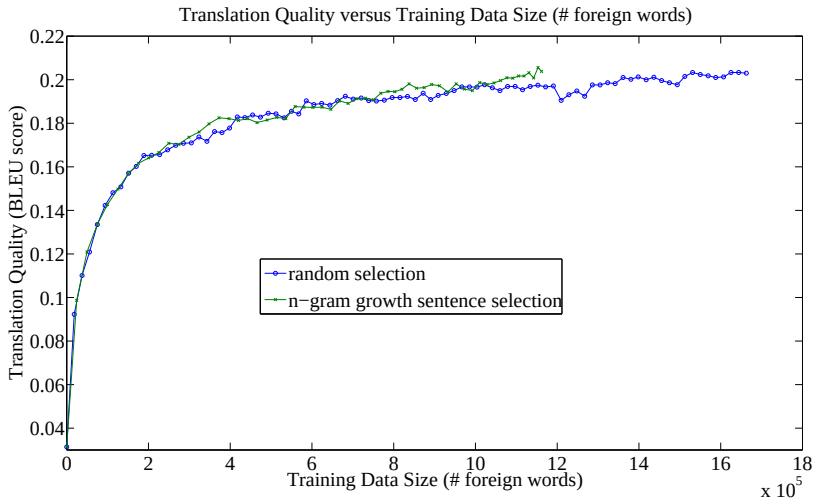


Figure 6.8: Performance of n-gram sentence selection versus random sentence selection with training data size measured by number of foreign words. The y-axis measures the performance of the Joshua MT system with a trigram language model.

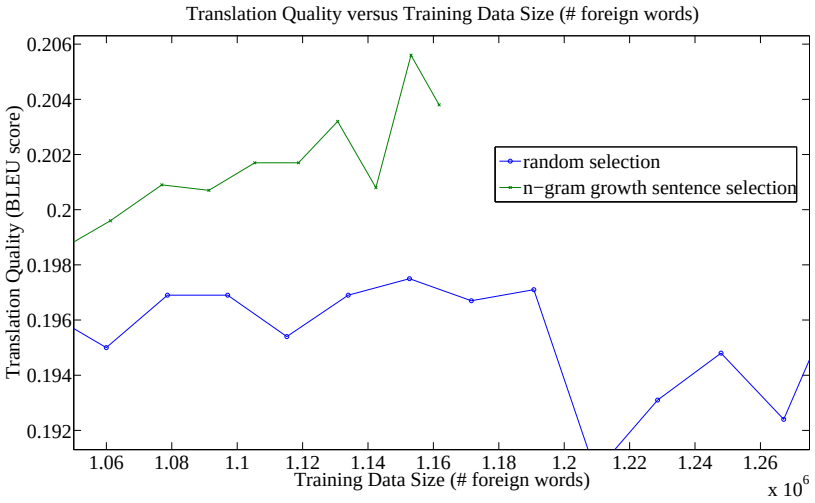


Figure 6.9: Performance of n-gram sentence selection versus random sentence selection with training data size measured by number of foreign words (close-up in the area where n-gram sentence selection stops requesting translations.

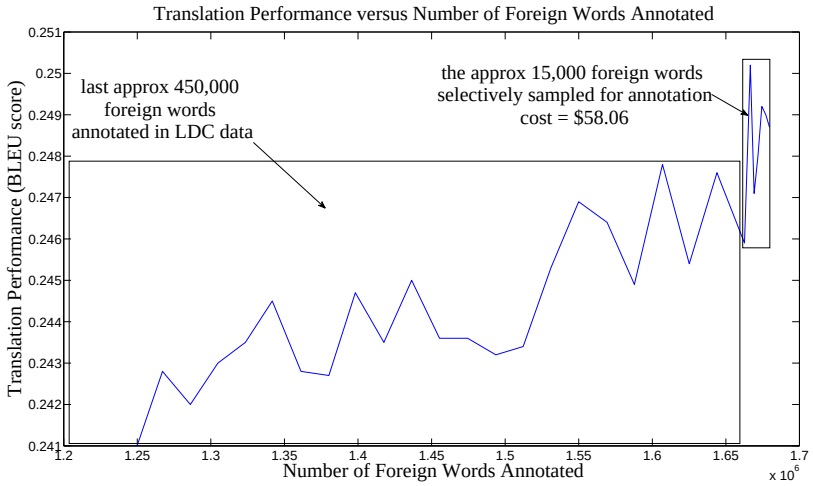


Figure 6.10: Performance of highlighted n-gram selection of additional translation from BBC web crawl data.

# times in test set	# new Urdu phrases
0	5624
1	152
2	24
4	2
5	2
6	1
10	1

Table 6.1: Coverage of new Urdu phrases in the (relatively small) test set.

Table 6.1 shows how often the new Urdu phrases occur in the test set. Notable is that 5624 out of the 5806 phrases for which we collected translations never occur in the test set. Since the test set is relatively small, it may be the case that it is understating the improvements in our translation system. We selected the order in which to ask for translations based on counts in the large unlabeled (i.e., monolingual) pool of BBC data we crawled. If one has knowledge of a particular domain in which the translation system will be tested, selecting the order in which to ask for translations based on counts obtained from monolingual text in that particular domain is a technique we expect will help to ensure that the solicited translations are likely to occur in the test data. The effectiveness of this domain adaptation strategy remains to be verified in future work.

Figure 6.11 shows an example of our strategy working. Here, the after system translates the Urdu word for Nagaland correctly, whereas the before system did not.

Figure 6.12 shows an example where the strategy is working partially but not as well as it might. The Urdu phrase means “gowned veil”. However, since the word aligner just aligns the word to “gowned”, we only see “gowned” in our output. This prompts a number of discussion points. First, the ‘after system’ has better translations but they’re not rewarded by BLEU scores because the references use the words ‘burqah’ or just ‘veil’ without ‘gowned’. Second, we hypothesize that we may be able to see improvements by overriding the automatic alignment software whenever we obtain a many-to-one or one-to-many (in terms

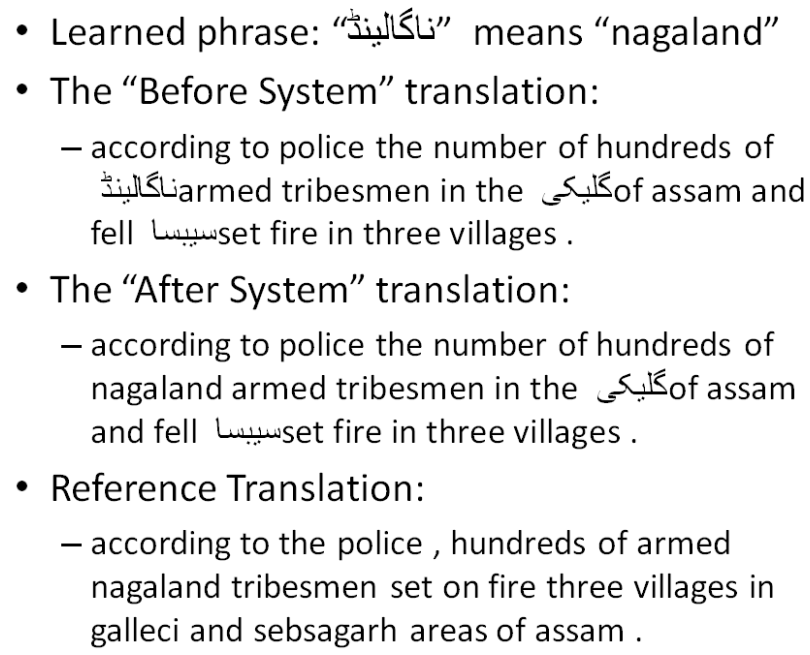


Figure 6.11: Example of strategy working.

of words) translation for one of our trigger phrases. In such cases, we'd like to make sure that every word on the 'many' side is aligned to the single word on the 'one' side. For example, we would force both 'gowned' and 'veil' to be aligned to the single Urdu word instead of allowing the automatic aligner to only align 'gowned'.

Figure 6.13 shows an example where our "before" system already got the translation correct without the need for the additional phrase translation. This is because though the "before" system had never seen the Urdu expression for "12 May", it had seen the Urdu words for "12" and "May" in isolation and was able to successfully compose them. An area of future work is to use the "before" system to determine such cases automatically and avoid asking humans to provide translations in such cases.

6.4 Conclusions and Future Work

A major bottleneck in the building of high-quality SMT systems is providing enough training data. Active learning approaches for selectively sampling data to try and cover relatively frequent n-grams that we don't have in our existing training data were attempted. These approaches showed promise for increasing the quality of SMT systems with only small amounts of additional training data.

It's important to measure annotation effort carefully and we showed how changing the measurement unit from number of sentences to number of foreign words can substantially change the conclusions one would draw. We posit that number of foreign words is more accurate than number of sentences because sentences could have different amounts of annotation effort required to translate them depending on how long and difficult the sentence is. Although previous AL studies have often indicated the value of selecting examples for annotation on which the current system is most uncertain, for SMT we posit that this might overwhelm the learner because the fundamental stage of word alignment might not get performed correctly. Motivated by these considerations, we proposed a selection algorithm which solicits translations for only

- Learned phrase: “برقعے” means “gowned veil”
- The “Before System” translation:
  - ' in برقعےmaulana in برقعےgeneral is not islam in ! برقعے
- The “After System” translation:
  - ' is in gowned , maulana gowned , in general is not islam in gowned !
- Reference translations:
  - in burqah , there is the maulana . in burqah , there is the general . in burqah , there is no islam .
  - molana is under veil , and the general is under veil , but islam is not under veil .

Figure 6.12: Example showing where we can improve our selection strategy.

- Example: “بارہ مئی” means “12 may”
  - The “Before System” translation:
    - all party conference in speeches during the meeting two days of karachi on 12 may in connection with the incidents , was strongly criticized the mqm .
  - The “After System” translation:
    - during the meeting two days of all party conference held in karachi on 12 may in the speeches about the incidents were strongly criticized the mqm .

Figure 6.13: Example showing where we can improve our selection strategy.

parts of sentences at a time. Combined with using Amazon Mechanical Turk, we were able to achieve translation gains with relatively small expenditures.

Analysis of our results revealed that there is room for future work by forcing word alignments in many-to-one and one-to-many situations to override the failures of automatic word aligners in such situations. Also, we can improve our n-gram selection algorithm by not soliciting translations when the current system can already translate the n-gram correctly even though it has never seen it before in its training data. In order to do this, we can develop a method for the system to self-predict whether it knows how to translate an n-gram or not.

Other areas of future work are to improve the translations gathered via AMT. Though we can obtain translations for low cost there, sometimes the translations are a bit sloppy. For example, they will put the two words "at" and "least" as one word in their translation: "atleast". This kind of training data can be harmful. We envision that using a post-editing task where English speakers are asked to correct spelling mistakes and other simple mistakes like "atleast" could improve the training data.

Chapter 7

Transliteration

One of the largest challenges of applying statistical translation techniques to low-resource conditions is a lack of training data. A novel text to be translated may contain many words which were never observed in the limited training corpus. Such words are often content-bearing words or names, containing critical information. Traditional statistical MT paradigms generally drop such words without attempting to translate them. While such omissions may not lead to catastrophic performance degradations under high-training conditions, they represent a severe challenge in low-resource environments.

In Urdu-English MT, we observed that 2% of words in a development set were out-of-vocabulary (OOV) with respect to our 1.8M-word training bitext. With the help of an annotator, we found that approximately 33% of these words were phonetically transliterable; for example, proper names or borrowed words. In Figure 7.1 below, we list a few examples of transliterable OOV words:

وکیپیڈیا	Wikipedia
کمیونیکیٹ	Communicate
پرشانتی	Prashanti

Figure 7.1: Examples of transliterable out-of-vocabulary words in Urdu test sets.

Because we observed many OOV words to be phonetically transliterable, we partially addressed the lexical coverage problem by developing a transliteration system and integrating it into the Joshua MT decoder.

7.1 Transliteration Model

In our system, transliteration is represented as a two-stage process. The first stage treats transliteration as a *monotone character translation* task, similar to the work of (Knight and Graehl, 1997). The second stage of our transliteration procedure aims to fix mistakes generated during the first-stage transliteration.

7.1.1 First-stage model: character translation

At training time, given a list of Urdu-English name pairs, we first perform character-to-character alignment using an off-the-shelf tool such as Giza++ or the Berkeley Word Aligner. Next, we find character-cluster pairs which conform to the alignment graph for a word pair; these are analogous to phrase pairs in statistical

MT. We build a table of such character cluster mappings, annotated with translation probabilities. Finally we extract a large, frequency-annotated name list from large English corpora, and use this to train a character language model prior. Having trained these components, we use them in conjunction with the off-the-shelf Joshua MT decoder as a first-pass transliterator.

During decoding, a novel Urdu word is segmented into character clusters, and each character cluster is translated to an English cluster. Unlike in the closely analogous process of phrasal machine translation, in phonetic transliteration the translated character clusters are never reordered. Transliteration hypotheses are scored using a log-linear model which makes use of character cluster translation scores and a character-LM prior score. The result of first-pass transliteration is a single English phonetic gloss of the Urdu word.

7.1.2 Second-stage model: post-editing

In the second stage of transliteration, a post-editor component rewrites the English hypothesis from the first pass to correct near-misses. The post-editor inserts missing vowels, which are often implicit in Urdu; it can also replace English syllables with phonetic matches or near-matches. For example, the post-editor has the ability to change a ”ff” character cluster into ”gh”.

We learn phonetically-matched English character cluster pairs by applying the technique of *pivoting*, which has previously been used in machine translation to learn models of English or foreign paraphrasing. In pivoting, we begin with a large, character-aligned name pair list, in which the target language is English but the source language need not be Urdu. In our experiments, we use an Arabic-English name pair list with approximately 7500 entries. We first extract English-to-Arabic and Arabic-to-English rule tables, and then compose the two tables to create an English-to-English table. The resulting table associates each English-to-English substitution with a probability score. We then augment this table with English vowel-insertion rules, as well as rules which simply copy the English input. The latter rules allow the post-editor the option to output the first-stage transliteration hypothesis unchanged. In Tables 7.1 and 7.2 below we show some examples of English-to-English subsitutions.

Input character cluster	Output character cluster
ch	sh
k	ck
ff	v
p	b
wa	oa

Table 7.1: Examples of English-to-English transfer rules learned via pivoting.

Input character cluster	Output character cluster
a	a
b	b
b	ba
b	be
b	bi

Table 7.2: Examples of English-to-English transfer rules generated by a simple script

Once we have constructed the English-to-English table, we run the Joshua decoder a second time to edit the first-pass transliteration output. This time, the Joshua decoder is run using the English-to-English rule

table and an English character-LM prior. We use minimum-error-rate training on a development set to tune the decoder's preference for editing, rather than simply copying, the first-pass output.

7.1.3 Semantically-targeted transliteration

We trained two transliteration systems, one for person names and one for all other semantic types (including non-names). These two systems shared all components except for the character LM and the dataset used for decoder weight tuning. For the person-name transliteration system, we trained our character language model from a large list of automatically extracted English person names. In the non-person-name transliteration system, we trained a model from a large list of English words, without regard to semantic type.

7.2 Data Gathering and Bootstrapping

7.2.1 Introduction

In order to train the transliteration module, we gathered pairs of names that were likely to be transliterations of one another. We obtained name pairs from three sources: the Urdu-English parallel corpus, Amazon's Mechanical Turk system, and Wikipedia.

7.2.2 Name pair extraction from Urdu-English parallel corpus

The first source of transliteration name pairs was the Urdu-English parallel corpus. We extracted name pairs from this corpus via a semi-automated bootstrapping procedure.

Initial seed data

At the outset of our work, no name pairs were available. Our first task was to develop a seed corpus to begin the process of bootstrapping. We first used the Phoenix English namefinder to find namespans in the Urdu-English bitext. We then projected each English namespan to Urdu using automatic word alignments obtained from Giza, in the process creating a list of English-Urdu name pair candidates. We assigned each name pair a *phonetic match score* based on a criterion similar to Soundex. By manually selecting a high score threshold that prioritized precision, we were able to extract a subset of approximately 200 high-confidence name pairs to serve as seed data.

Parallel text bootstrapping

After extracting the seed training data, we trained a first-pass transliteration model. We used this model to attempt to transliterate every Urdu word in the 1.8M-word Urdu-English bitext. If the attempted transliteration of an Urdu word had low edit distance to a word in the corresponding English sentence, we extracted the resulting word pair as a candidate for transliteration training. A bilingual annotator was able to quickly remove incorrect candidates from the list, resulting in a final list of approximately 600 correct person name pairs, as well as approximately 300 correctly transliterated non-person-name word pairs.

7.2.3 Mechanical Turk annotation

In addition to seed data generation and bootstrapping, we collected thousands of additional name pairs using the Mechanical Turk system.

To elicit transliteration pairs from Mechanical Turk workers, we first gathered very large Urdu and English monolingual name lists by tagging our very large corpora of English and Urdu BBC data with

Resource type	Source	Amount of Data	Purpose
Person name pairs	Parallel corpus Mechanical Turk Wikipedia **Total**	800 word pairs 12,380 word pairs 850 word pairs **14,030 word pairs**	Training, tuning for first-pass person-name transliteration system
Non-person word pairs	Parallel corpus	300 word pairs	Tuning for first-pass non-person-name transliteration system
English person name list, with frequencies	English Gigaword corpus, processed by Phoenix namefinder	90 million name instances	Character LM training for person-name transliterator
English non-person word list, with frequencies	English Gigaword corpus	>2.5B words	Character LM training for non-person-name transliterator

Table 7.3: Data used for transliteration

the Phoenix name tagger. This yielded 22,565 English names and 27,449 Urdu names. We decided to have Mechanical Turk workers transliterate several thousand of the frequent, but not most frequent, English names into Urdu and Urdu names into English. That is, we gathered transliterations for the names that were tagged by Phoenix most frequently after the 500 most frequent names were discarded. We did not want Mechanical Turk workers to produce only transliterations that they had seen in print before, but rather we wanted to gather their own transliteration instincts by presenting them with names that they were less likely to have seen. Additionally, those words that were tagged very infrequently as names were often not names at all, and, thus, less likely to be naturally transliterated. So, gathering transliterations for this biased name list eliminated the potential for several sources of noise. We posted 5,470 names from each monolingual name list, and, in most cases, we had three Mechanical Turk workers give transliterations for each name. We approved an average of 2.1 transliterations per English name and 2.7 transliterations per Urdu name. We did not approve answers that were obvious instances of cheating (use of an online translation system, for example) or that were left blank. Our final corpus included 12,384 unique pairs of unigrams for the training and development of the transliteration module.

7.2.4 Mining name pairs from Wikipedia

In addition to the above data gathering techniques, we also crawled the English and Urdu Wikipedia sites and gathered name pairs from the titles of entries about people that had corresponding Urdu and English pages. There were 890 such pages, which yielded 852 unique name pairs.

7.3 Summary of transliteration resources

The complete specification of the data available for each transliteration model is given in Table 7.3.

7.4 Intrinsic evaluation of transliteration quality

7.4.1 Setup

In order to test the performance of the transliteration module directly (rather than only by the end-to-end translation task), we created a test set of 1000 name pairs. Half of these pairs were taken from the English to

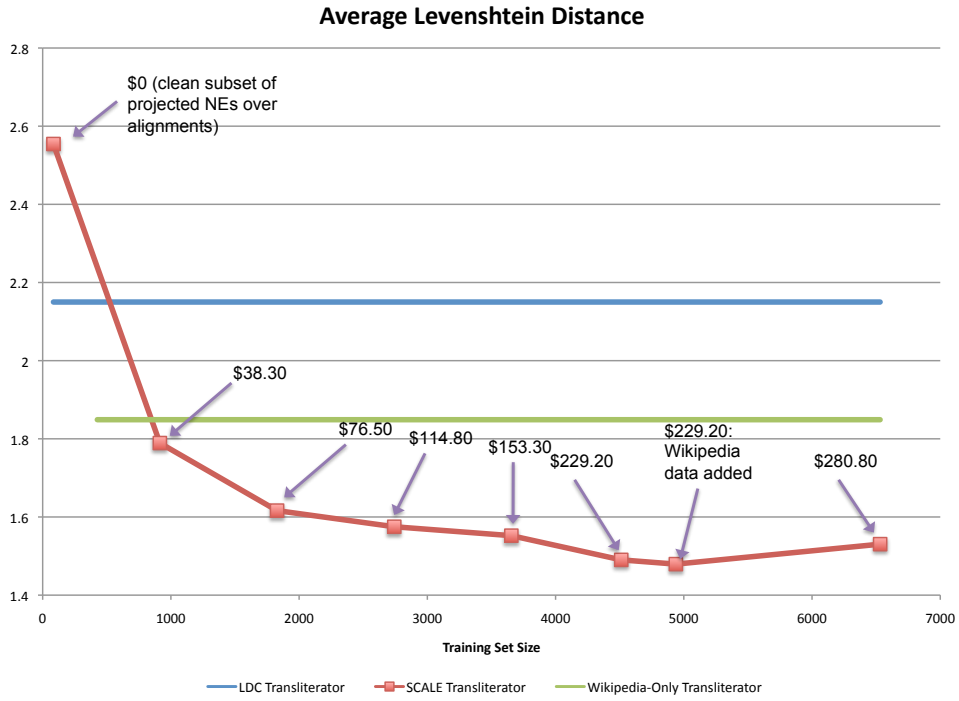


Figure 7.2: Average Levenshtein distance between transliterator output and test set gold standards.

Urdu Mechanical Turk data and half from the Urdu to English Mechanical Turk data. For all pairs, at least two of three Mechanical Turk workers gave the exact same transliteration for a name. We also eliminated pairs for which the edit distance between the romanized Urdu (again using an extension of the Buckwalter transliteration map) and the English was large. That is, we did not include pairs that were obviously not transliterations. Thus, the final set of 920 pairs is very clean and represents names for which there is a transliteration consensus. There was no overlap between either side (English or Urdu) of the testing set pairs and the training and development sets.

7.4.2 Results

Figure 7.2 shows the average Levenshtein Edit Distance between the transliteration module output and the test set gold standard transliterations as the module was trained on various amounts of training data. It should be noted that the training set size and the development set size were equal. So, the second point on the graph refers to the module built with about 900 pairs for training and about 900 pairs for development. This graph also notes the performance of the baseline LDC transliterator, which was included in the Urdu language pack. It also shows the performance of the transliteration module when trained on only the freely available Wikipedia name pairs. For each point on the SCALE transliterator performance line, the cost of gathering the name pairs from Mechanical Turk is noted. The first point on the graph uses only those name pairs gathered by projecting names tagged on the English side of the parallel corpus over the alignments. It is easy to see that our method for using both Wikipedia name pairs and the name pairs gathered from Mechanical Turk workers (at a low cost) hugely outperforms our two baseline transliterators.

Figure 7.3 is another performance graph. Similar to Figure 7.2, this graph shows a count of the number

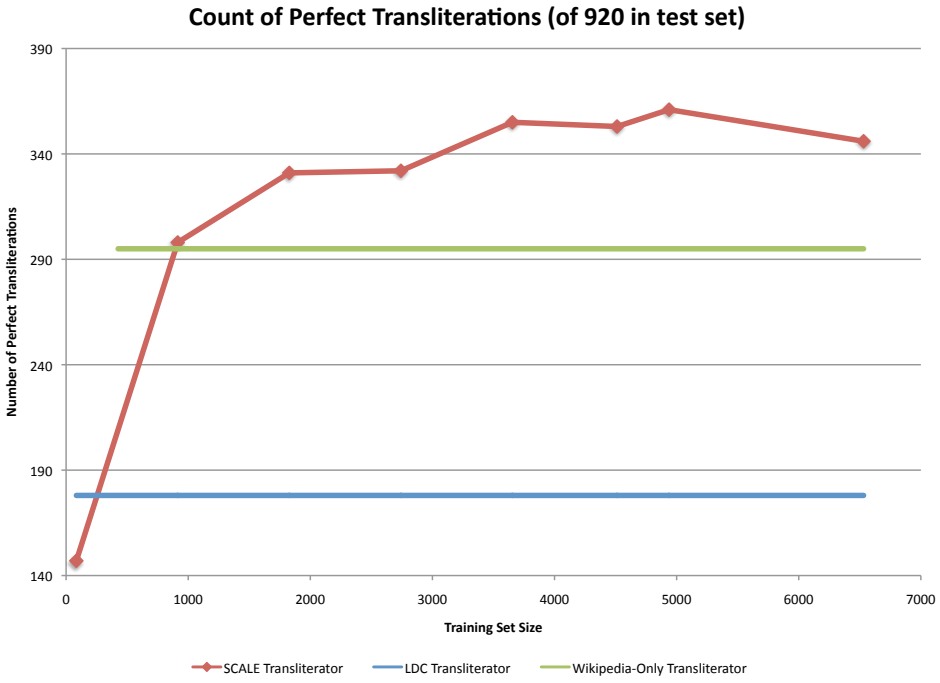


Figure 7.3: Count of perfect transliterations according to the test set gold standards in output of SCALE transliterator

of perfect transliterations produced by the module. The trends are the same.

7.4.3 Sample transliterator output

Table 7.4 shows some examples of the output produced by the SCALE transliterator as it is trained on various numbers of name pairs. It is nice to see that the output quality generally increases as more training pairs are used. The examples in Table 7.5 are indicative of the oftentimes subjective nature of the transliteration task. Frequently, more than one transliteration is probably acceptable to most readers. We estimate that about half of the 'incorrect' transliteration outputs look like those in Table 2. That is, they are probably reasonable, and certainly readable, output. This is especially true when transliterating a native Urdu name into English or a native English name into Urdu. This is important to keep in mind because translation metrics like Bleu will give no credit to near-perfect name transliteration output, even though producing translations that include these names will likely greatly increase the readability of the output.

7.4.4 Impact of Post-editing on Transliteration Performance

We assessed the impact of the post-editor component on transliteration performance both quantitatively and qualitatively. Quantitatively, we investigated the impact of post-editing on the transliterator's performance on a development set, as measured by Character Error Rate (CER). We also performed anecdotal qualitative investigations by manually inspecting transliterator output before and after post-editing.

Both qualitatively and quantitatively, we found that post-editing was of unclear utility in person-name transliteration, but of significant utility in non-person-name transliteration. In Figure 7.4, we show examples

Training Data Size	Example 1	Example 2	Example 3
84	orosco	hu	moqtaders
914	urska	yao	muqtadera
1828	wersk	yao	muqtra
2742	urska	yao	muqtara
3655	versk	yao	muqtadera
4512	orsik	yaho	muqtadera
4938	versk	yaho	muqtadra
6531	warsak	yahu	moqtadar
Correct Transliteration	warsac	yahoo	muqtadra

Table 7.4: Examples of transliteration output

Transliterator Output	Correct (Reference) Answer
majidi	majeedi
kanaan	kanan
mahmud	mohmmad
hamdallah	hamdullah
calia	kalia
alan	allen
nawaz	nawaaz
mediha	madieha
pirzada	pir-zada
akshay	akhshay
biberg	biburg
sethi	sethy

Table 7.5: Examples of reasonable incorrect output

Transliteration System Type	CER (Before post-editing)	CER (After post-editing)
Person-names	.201	.187
Non-person-names	.199	.110

Table 7.6: Quantitative measurement of improvement from post-editing, measured on a non-held-out development set

of the impact of post-editing in non-person-name transliteration. These examples were drawn from the NIST MT09 evaluation set, a blind data set which we also used for MT evaluation. In Table 7.6, we show the quantitative impact of post-editing for both types of transliteration. Because of limited data availability, we measured quantitative improvements on a non-held-out development set only. For the person-name transliteration model, 700 name pairs were used as development data, while for the non-person-name model, 300 word pairs were used. Additionally, it should be noted that these experiments were conducted earlier in the development cycle, and therefore not all training data was available. In particular, the system was trained using 1400 person name pairs of training data.

Name	First-pass	Post-edited	Reference
پاکستان	Pakstan	Pakistan	Pakistan
یونیورسٹی	yuniorsty	university	university
اسمبلی	asmbly	assembly	assembly
انٹیلجنس	antilgns	intelligence	intelligence
سیکرٹری	sakrtre	secretary	secretary
ڈیزائین	dizayeen	dizayeen	design
کمانڈر	kamander	commander	Commander
ایڈیشنل	aidesnl	desnl	additional

Figure 7.4: Examples (taken from blind NIST09 test) of post-editing in non-name transliteration

7.5 Joshua MT decoder integration

After developing a transliteration system for person-names and one for non-person-names according to the above methods, we integrated these systems into the Joshua MT decoder.

7.5.1 Generating N-best transliteration hypotheses

Given a novel Urdu test sentence to be translated, we first select a subset of words as candidates for transliteration. In particular, we identify all Urdu words which are either out-of-vocabulary or low-frequency (using a threshold of 5 occurrences) with respect to the Urdu-English bitext. We then assign a name type (including a category for non-names) to each transliteration candidate using the Phoenix Urdu name tagger. Next, as discussed in section 7.1.3, we dispatch each transliteration candidate to either the person-name or non-person-name transliteration system, depending on the Phoenix-assigned name type. For each Urdu word, this process yields an N-best list of transliteration options. We truncate this list to N=20 options, and we

Feature name	Description
IsTranslit	Binary feature set to 1 for all transliteration rules
IsName	Binary feature set to 1 for all attempted transliterations of words Phoenix found as names.
LogBitextSourceFrequency	The log of the source frequency in the bitext. Penalizes attempted transliteration of more frequent words.
NormalizedTranslitMargin	Zero for all transliteration options except the 1-best. For the 1-best, this is the difference in transliterator scores between the 1st and 2nd best hypotheses, divided by the character length of the Urdu word. Gives bonus to high-margin transliteration options.
LogTargetWordModelProb	The log of the target word probability, according to a semantically appropriate English word model.
NormalizedCharTransducer	Length-normalized character transducer log probability.

Table 7.7: Full list of transliteration-specific MT rule features.

also remove any options from the N-best list which have very low unigram word probability, according to a semantically appropriate word model.

7.5.2 Creating translation rules

We add the filtered lists of transliteration options to Joshua's chart as alternate translation rules. As part of this process, we augment these translation rules with nonterminal information as well as feature vectors.

Semantically appropriate nonterminals

According to Joshua's grammar-based translation formalism, every translation rule must be covered by a nonterminal. Rather than choosing a generic nonterminal such as 'X', we wish to assign a syntactically and semantically appropriate nonterminal label, to help ensure that nearby structures in the sentence can be decoded in a syntactically coherent way. We accomplish this as follows: in a process run before decoding for each semantic type, we generate a list of the 5 most common part-of-speech labels assigned to low-frequency words of the given semantic category. Since the translation grammar makes use of semantic grafting as described in Chapter 2, these part-of-speech tags will carry explicit semantic information. As an example, we found the most common semantically-augmented part-of-speech tags for low-frequency words in the PERSON name category to be {NNP-PERSON, NN-PERSON, NNPS-PERSON, JJ-PERSON, NNS-PERSON}. Once we have determined such a list of allowable nonterminals for each semantic type, we simply allow any permissible nonterminal to cover any transliteration candidate.

Transliteration-specific rule features

Joshua also requires every translation rule to be annotated with a vector of features. We assign transliteration rules a feature value of zero for all standard Joshua rule features (such as the relative-frequency feature $P(Target \mid Source)$), and add a collection of new features specific to transliteration hypotheses. The transliteration-specific features include a rough measure of confidence, a bonus for transliterating less-frequent source words, and a character-transducer-based transliteration coherence model. The complete list of features is given in Table 7.7.

7.5.3 Impact of transliterator integration

We tested the impact of transliterator integration in a small number of blind submissions to the NIST MT09 Urdu-English evaluation. We integrated the transliterator into the best Joshua system available, namely the semantically-grafted, syntax-aware system. In one submission, we transliterated all low-frequency words, while in a second submission we transliterated only low-frequency person names. Our baseline for comparison was the identical SAMT-grammar Joshua system without transliterations.

We compared the baseline against the transliteration-aware systems both quantitatively and qualitatively. In a quantitative comparison, whose results are shown in Table 7.8, transliteration yielded a small but notable improvement according to the automatic BLEU metric. As shown in the figure, transliterating words of all semantic types yielded slightly better performance than transliterating only words marked as person names.

Joshua Translation System	**NIST MT09 BLEU Score**
No Transliteration	.2958
Transliterate names only	.2980
Transliterate all types	.3010

Table 7.8: Impact of transliteration on BLEU metric in submissions to NIST MT09 evaluation.

We also qualitatively compared the best transliteration-aware system with the baseline system via manual inspection of decoder output. As expected, some sentences showed clear improvement via the improved lexical coverage allowed by the transliteration model, while other sentences showed little benefit. In some sentences, the transliteration model hypothesized incorrect transliterations for out-of-vocabulary words. More effectively filtering such incorrect translation options, such as through a more developed measure of confidence, is a potential avenue for future work.

Tables 7.9 through 7.11 show examples of Joshua decoder output with and without the transliteration feature.

Without Transliteration	**[UNKNOWN]** members said that the economic plan, expensive, and will not be effective.
With Transliteration	**Republican** members said that the economic plan, expensive, and will not be effective.
Reference	the republican members said that this economic plan is very “expensive” and will not be effective .

Table 7.9: Example of improvements from transliteration.

7.6 Conclusion

In our work during the SCALE workshop, we demonstrated the potential of a transliteration subsystem to improve machine translation in a low-resource training scenario. We observed this improvement both with automatic metrics and when manually investigating decoder output. Future work could aim to further build on these gains, by sharpening transliteration models, improving decoder integration models, and supplementing the MT decoder with additional contextual features (such as topic models or syntactic prior models) to better disambiguate the proper transliteration of a word given its sentence and document context.

Without Transliteration	in southern germany **[UNKNOWN]** a resident of the area of **[UNKNOWN]** **[UNKNOWN]** has left behind a big business group
With Transliteration	a resident of the area of **cuba** in south germany **adolf merkel** has left behind a big business group
Reference	**adolf merckle** of southern germany 's **swabia** area has left a large business group behind

Table 7.10: Impact of transliteration. Note that the location name "Swabia" was incorrectly transliterated to "Cuba." This example indicates the future room for improvement, e.g. from more explicit phonetic modelling.

Without Transliteration	however , **[UNKNOWN]** said that he **[UNKNOWN]** president to respect their age and are also due to yell at them , but they were saying truth from miles away .
With Transliteration	"however , **erdogan** said that he respects the **israeli** president and his age as a result of which they yell at them , but they were saying the truth from miles away .
Reference	however , later **erdogan** said that he respects **israeli** president and his age as well which is why he did not yell at him but whatever he was saying was miles away from truth .

Table 7.11: Example of improvements from transliteration.

Chapter 8

Semantic Annotation and Automatic Tagging

The goal of annotation and tagging is to add labels to text or structured text. When the labels are added by humans, we will refer to the process as annotation, and when the labels are automatically added, we refer to it as tagging. The examples shown in Figures 8.1 and 8.2 show tags for named entities and modality, respectively. Note that the modality tags are in pairs of triggers and targets, as will be explained below.

Below we will examine our representational assumptions for Named Entities and Modality. We will also examine our approaches to automatic annotation of modality and evaluation of our results.

8.1 Named Entities

8.1.1 Named Entity Annotation and Tagging

Urdu data was annotated with named entity labels under the government's REFLEX effort. The inventory of named entity tags for Urdu was a simplified subset of the ACE tag set. Table 8.1.1 below shows the Urdu and English tags along with exmples. The Phoenix IDF (sponsor tagger) was trained on the annotated data and was used for both English and Urdu tagging.

8.1.2 Integration of named entity tags with syntax

The goal of entity tagging for this project is to augment the node labels of syntax trees. The trees are used by Joshua and Cunei to learn generalizations and rules for machine translation. A tag percolation program

Input: The Central minister Vijay Kumar Malhotra has won from South Delhi seat.
Output: The Central <OCCUPATION minister> <PERSON Vijay Kumar Malhotra> has won from <GPE South Delhi> seat.

Input: Australian women cricket team's captain Obanda Clarke is my favorite player.
Output: <GPE-ite Australian> women cricket team's <OCCUPATION captain> <PERSON Obanda Clarke> is my favorite player.

Figure 8.1: Example of I/O for Named-Entity Tagging

Input: Americans should know that we can not hand over Dr. Khan to them.
Output: Americans <TrigRequire should> <TargRequire know> that we <TrigAble can> <TrigNegation not> <TargNOTAble hand> over Dr. Khan to them.

Input: He managed to hold general elections in the year 2002, but he can not be ignorant of the fact that the world at large did not accept these elections.
Output: He <TrigSucceed managed> to <TargSucceed hold> general elections in the year 2002, but he <TrigAble can> <TrigNegation not> <TargNOTAble be> ignorant of the fact that the world at large did <TrigNegation not> <TrigBelief accept> these <TargBelief elections>.

Figure 8.2: Example of I/O for Modality Tagging

English tag	Urdu tag	Example
AGE		50 years old
DATE	DATE	September 26, 2009
FACILITY		
GPE	LOCATION	New York
GPE–ite	LOCATION	Australian
LOCATION	LOCATION	
MONEY		
OCCUPATION		
ORGANIZATION	ORGANIZATION	
ORGANIZATION–ite	ORGANIZATION	
PERCENT		
PERSON	PERSON	
PERSON–NN	PERSON	
TIME	TIME	in the afternoon
UNK	UNK	
	TITLE	Mr.

Table 8.1: **Named Entity Tags for English and Urdu**

merges tags with phrase structure trees. The tags are passed up from lexical items like nouns to phrasal nodes like noun phrases. Figures 8.3 and 8.4 illustrate the process of merging named entity tagger output with phrase strucure trees. Figure 8.5 shows a syntax tree with named entity labels.

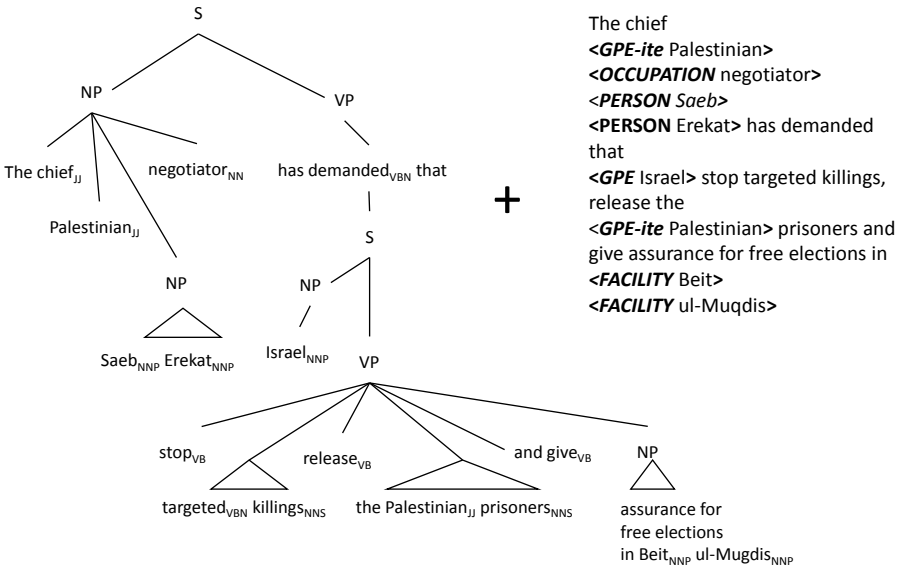


Figure 8.3: Input to Tag Percolation

8.2 Modality

Modality is an extra-propositional component of meaning. That is, it says something beyond the basic proposition. In *John may go to NY*, the basic proposition is *John go to NY* and word *may* indicates modality. The core cases of modality are related to possibility, necessity, permission, and obligation. Van der Auwera and Amman (Auwera and Ammann, 2005) define the core cases of modality as shown in Figure 8.2. Epistemic modality is about whether events are possible or whether they must necessarily be true. Deontic modality is about permission and obligation.

Many semanticists (Kratzer, 2009; von Fintel and Iatridou, 2009) define modality as quantification over possible worlds. *John might go* means that there exist some possible worlds in which John goes, whereas *John has to go* means that in all possible worlds (defined by some context) John goes. Another view of modality relates more to a speaker's attitude toward a proposition (Nirenburg and McShane, 2008). In this project, modality was defined broadly to include several types of attitudes that a speaker might have toward an event or state. A modality coding manual (Dorr et al., 2009) was produced before the SCALE began including the following modalities:

- **Belief**: with what strength does H believe P?
- **Requirement:** does H require P?

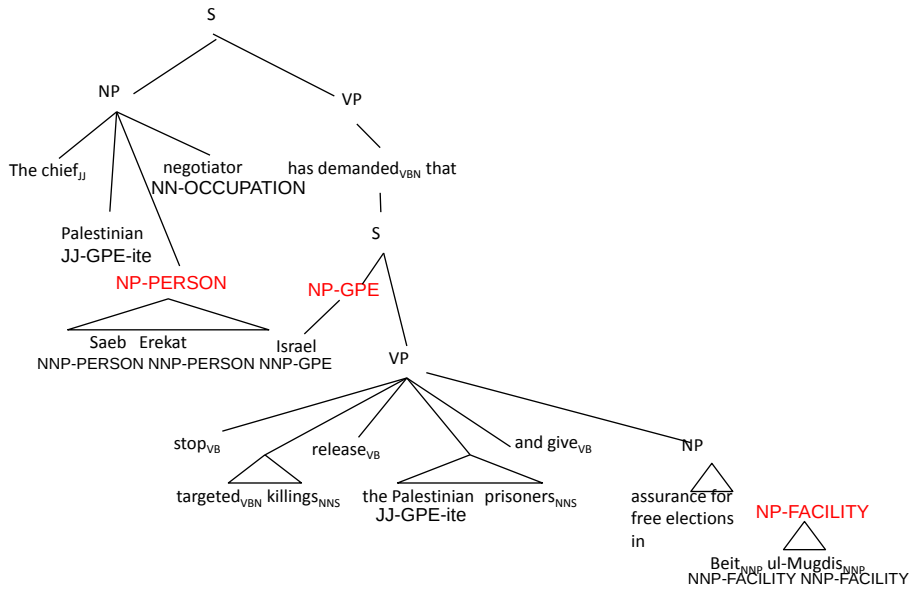


Figure 8.4: Named Entities Combined with Syntax

- **Permissive:** does H allow P?
- **Intention:** does H intend P?
- **Effort:** does H try to do P?
- **Ability:** can H do P?
- **Success:** does H succeed in P?
- **Want:** does H want P?

We also included negation of the modality itself or of the clause it scopes over. However, there are some concepts related to speaker attitude that we did not cover such as evidentiality (first hand or second hand knowledge of the proposition) and positive and negative sentiment (liking and hating).

Below we will explain how we dealt with the interaction of modalities with each other and with negation in the coding manual. The interaction is complex and is the topic of many papers in linguistic theory. However, for our task we had to create a simplified operational procedure that could be followed by language experts.

8.2.1 The anatomy of modality in sentences

In sentences that express modality, we identify three components, a trigger, a target, and a holder. The trigger is the word or string of words that expresses modality. The target is the event, state, or relation that the modality scopes over. The holder is the experiencer or cognizer of the modality.

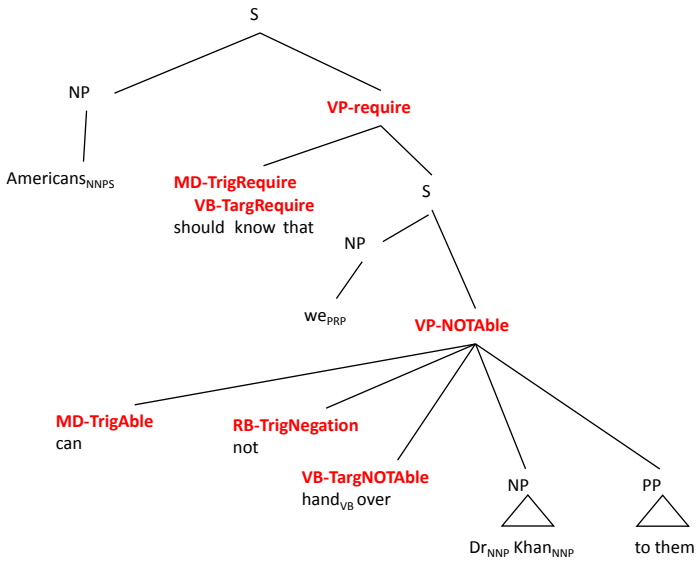


Figure 8.5: Modality Combined with Syntax

	Necessity	**Possibility**
Epistemic	John must go to NY	John might go to NY
Deontic or Situational	John has to leave now	John may leave now

Figure 8.6: **Core Cases of Modality (Van der Auwera and Amman)**

Modality Trigger: The trigger can be a word such as *should*, *try*, *able*, *likely*, or *want*. It can also be a negative element such as *not* or *n't*. Often, modality is expressed without a lexical trigger. For a typical declarative sentence (e.g., *John went to NY*), the default modality is strong belief when no lexical trigger is present. Modality can also be expressed constructionally. For example, obligation can be expressed in Urdu with a dative subject and infinitive verb.

8.2.2 The Modality Coding Scheme

Annotation of modality in Urdu began before the start of the SCALE. In preparation for this, we created a practical annotation scheme for modality. The annotation scheme needed to be implementable in the Callisto annotation interface, and it also had to be operationalized in a way that was easy and fast for Urdu language experts. Later in the project, the same annotation scheme was used for automatic tagging of modality in English. The annotation manual describes the labeling of target words only. For a word that is the target of a modality, the annotator chooses from thirteen modalities.

- H requires [P to be true/false]
- H permits [P to be true/false]

- H succeeds in [making P true/false]
- H does not succeed in [making P true/false]
- H is trying [to make P true/false]
- H is not trying [to make P true/false]
- H intends [to make P true/false]
- H does not intend [to make P true/false]
- H is able [to make P true/false]
- H is not able [to make P true/false]
- H wants [P to be true/false]
- H firmly believes [P is true/false]
- H believes [P may be true/false]

Lingusitic simplifications/efficient operationalization Four linguistic simplifications were made for the sake of efficient operationalization of the annotation task. The first linguistic simplification deals with the scope of modality and negation. The first sentence below indicates scope of modality over negation. The second indicates scope of negation over modality:

```
He tried not to criticize the president.
He didn't try to criticize the president.
```

The interaction of modality with negation is complex to explain, but was operationalized easily. First consider the case where negation scopes over modality. Four of the fourteen modalities are composites of negation scoping over modality. For example, the annotators can choose *try* or *not try* as two separate modalities. Five modalities do not have a negated form. This is because they are often transparent to negation. For example, *I do not beleive that he left* sometimes means the same as *I beleive he didn't leave*. Merging the two is obviously a simplification, but it saves the annotators from having to make a difficult decision.

After the annotator chooses the modality, the scoping of modality over negation takes place as a second decision. For example, for the sentence *John tried not to go to NY*, the annotator first identifies *go* as the target of a modality. Then the annotator chooses *try* as the modality. And finally, the annotator chooses *false* as the polarity of the target.

The second linguistic simplification is related to a duality in meaning between *require* and *permit*. Not requiring P to be true is similar in meaning to permitting P to be false. Therefore, annotators were instructed to label *not require P to be true* as *Permit P to be false*. Conversely, *not Permit P to be true* was labeled as *Require P to be false*.

The third simplification relates to entailments between modalities. Many words have complex meanings that include components of more than one modality. For example, if you manage to do something, you tried to do it and you probably wanted to do it. In order to help annotators decide which modality to choose in these cases, the modalities are ordered. The annotator chooses the first one that applies.

The final linguistic simplification is that we did not require annotators to mark nested modalities. For a sentence like *He might be able to go to NY* only ability is marked for the target word *go*. The epistemic

modality expressed by *might* is not annotated. We made this decision because of time limits on the annotation task. We did not think that annotators would have time to deal with syntactic scoping of modalities over other modalities.

8.3 Automatic Annotation of Modality

A modality tagger produces text or structured text in which modality triggers and/or targets are identified. This section describes four modality taggers: a string-based English tagger, a structure-based English tagger, an Urdu tagger trained with Phoenix, and an English tagger trained with Phoenix. We will first describe a modality lexicon that is shared by the string-based English tagger and the structure-based English tagger.

8.3.1 The English modality lexicon

The modality lexicon is a list of modality trigger words (Dorr, Filardo, Bloodgood, Piatko). Appendix A provides the full listing. Each entry in the modality lexicon consists of:

- A string of one or more words: for example, *should* or *have need of*. The words are inflected, so *require*, *required*, *requires*, and *requiring* are separate entries in the modality lexicon.
- A part of speech for each word: The part of speech helps us avoid irrelevant homophones such as the noun *can*.
- A modality: one of the fourteen modalities described above, further specified according to whether Negation is applicable. Specifically, the Modality classes included Require, Permit, NotPermit, Succeed, NotSucceed, Effort, NotEffort, Intend, Able, NotAble, Want, NotWant, FirmBelief, Belief, NotBelief, and Negation. Note: Not all words were implemented for each of these cases.
- A head word: the primary phrasal constituent to cover cases where an entry is a multi-word unit, e.g., the word *hope* in *hope for*.
- One or subcategorization codes: the codes are derived from syntactic codes provided in the *Longman's Dictionary of Contemporary English* (LDOCE). They include transitive verb, intransitive verb, transitive/intransitive verb plus infinitive verb phrase, etc. They are described in more detail below. The subcategorization codes indicate which syntactic templates are used by the structure-based tagger.

Appendix B provides the rules used to map between LDOCE codes and subcategorization codes, e.g., the appearance of LDOCE code T3 or D3 or V3 indicates that the two templates V3-passive-basic and V3-I3-basic are applicable.

We note that most intransitive LDOCE codes were not applicable to modality constructions. For example, *hunger* (in the *Want* modality class) has a modal reading of "desire" when combined with the preposition *for* (as in *she hungered for a promotion*), but not in its pure intransitive form (e.g., *he hungered all night*). Thus the LDOCE code `I` associated with the verb *hunger* was hand-changed to `I-FOR` for our purposes. There were 43 such exceptions, as listed in the second part of Appendix B. Once the LDOCE codes were hand-verified (and modified accordingly), the mapping to subcategorization codes was applied.
The modality lexicon entry for the verb *need* is shown here:

need

- **Pos:** VB

- **Modality:** Require
- **Trigger word:** Need
- **Subcategorization codes:**
  - **V3-passive-basic**
    The government is needed to buy tents.
  - **V3-I3-basic**
    The government will need to work continuously for at least a year.
    We will need them to work continuously.
  - **T1-monotransitive-for-V3-verbs**
    We need a Sir Sayyed again to maintain this sentiment.
  - **T1-passive-for-V3-verb**
    Tents are needed.
  - **modal-auxiliary-basic**
    He need not go.

8.3.2 The string-based English modality tagger

The string based tagger operates on text that has been tagged with parts of speech by the Basis parser. The tagger marks spans of words/phrases that exactly match modality trigger words in the modality lexicon described above and in Appendix A, and that exactly match the same parts of speech. This tagger identifies the target of each modality using the heuristic of tagging the next non-auxiliary verb to the right of the trigger.

Spans of words can be tagged by this tagger multiple times with different types of triggers and targets. For rule-based machine translation experiments, only a single type of tag was allowed per word. The grafting process of adding modality labels, as outlined in Section 2.3, used a heuristic to prioritize and pick a single tag for each word.

8.3.3 The structure-based English modality tagger

The structure-based tagger operates on text that has been parsed with the Basis parser. We used a version of the parser that produces flattened trees. In particular the flattener deletes VP nodes that are immediately dominated by VP and NP nodes that are immediately dominated by PP or NP. The parsed sentences are processed by T-Surgeon rules. Each T-surgeon rule consists of a pattern and an action. The pattern matches part of a parse tree and the action alters the parse tree. More specifically, the pattern finds a modality trigger word and its target and the action inserts tags like `TrigRequire` and `TargRequire` for triggers and targets for the modality Require.

The T-surgeon patterns are automatically generated from a set of templates and the verb class codes from the modality lexicon. The sentences below are representative of the templates. The target words are italicized.

- allow them *some luxuries*
- believe that it is *raining*
- *The plan* flopped
- failed to *provide* aid

- failed in its *plan*
- succeed in *rescuing* refugees
- successful as a *mother*
- successful in their *efforts*
- likely to *succeed*
- The *plan* was successful.
- the successful *plan*
- seem *happy*
- the need for *tents*
- the need to *fight*
- want *tents*
- *Tents* are needed.
- The insurgents are required to *withdraw*.

8.3.4 The Urdu Phoenix Modality Tagger

Urdu modality tagging was done by using Berkeley word alignment on parallel English and Urdu texts, projecting English modality tags (from one or the other of our English tagging approaches) to Urdu, building a Phoenix model on the projected tags, and applying the model to novel Urdu text.

8.3.5 The English Phoenix Modality Tagger

Although we already proposed two methods for automatic English modality tagging, we also built Phoenix models to tag English modality using the automatically tagged English as the training data. The purpose was to gauge whether the automatic English modality tagging approaches produced training data good enough to learn from in a single-language context.

We applied both English modality tagging approaches to the SCALE English training data and the English dev data. We projected the trigger and target tags from the English to the Urdu data for both training and dev data. Phoenix models were built using the English and Urdu training data. These models were used to tag the English and Urdu dev data. We evaluated whether the automatic English tags and projected Urdu tags were good enough to train Phoenix models by treating the directly tagged English and projected Urdu on the dev data as truth, and calculating precision and recall on the model-tagged dev data.

8.3.6 Evaluation of Modality Work

We automatically tagged terms for modality in English and Urdu. We identified two categories of modality words. Triggers are words that indicate the presence of modality semantics, such as "never," and "ought." Targets are the words to which the modality applies. For example, in "John never lived in New York," the target of "never" is "lived." English modality tagging for triggers was done by two different methods. In the first approach, triggers were identified from a word-list, and targets were identified as the next verb. In the second approach, we wrote syntactic rules using Tsurgeon to identify triggers and targets. Urdu modality tagging was done by using Berkeley word alignment on parallel English and Urdu texts, projecting English modality tags (from one or the other of our English tagging approaches) to Urdu, building a Phoenix model on the projected tags, and applying the model to novel Urdu text. Although we already proposed two methods for automatic English modality tagging, we also built Phoenix models to tag English modality

using the automatically tagged English as the training data. The purpose was to gauge whether the automatic English modality tagging approaches produced training data good enough to learn from in a single-language context.

We applied both English modality tagging approaches to the SCALE English training data and the English dev data. We projected the trigger and target tags from the English to the Urdu data for both training and dev data. Phoenix models were built using the English and Urdu training data. These models were used to tag the English and Urdu dev data. We evaluated whether the automatic English tags and projected Urdu tags were good enough to train Phoenix models by treating the directly tagged English and projected Urdu on the dev data as truth, and calculating precision and recall on the model-tagged dev data.

Phoenix was trained on the following four training corpora: tsurgeon applied to English, lookup applied to English, tsurgeon applied to English and projected to Urdu, and lookup applied to English and projected to Urdu. We used these models to tag the development data. Recall that there are four English development files and one Urdu development file, and that the four English development files are distinct translations of the Urdu. We compared these Phoenix-tagged development files against files that were tagged directly by tsurgeon and lookup. For the English gold standard development set tags, we simply used each of these two direct models to tag each of the four reference translation sets. For the Urdu gold standard development set tags, we projected the modality tags on each of the four English development files to the single Urdu file. This resulted in four distinct gold standard Urdu modality-tagged development files for each model.

We consider the results of the four Phoenix tagging models to be the candidate systems, and the corresponding directly automatically tagged documents (or projection of the direct automatic tags to the Urdu documents) to be the references. The goal of the evaluations is the gauge informally whether the automatic modality taggers provide enough information to the Phoenix trainer that those models can even predict what the automatic tagger would do on new data. The hope is, of course, that they would not only be able to replicate the tags that tsurgeon and lookup would produce but also generalize patterns and produce additional good tags. Furthermore, in the absence of a bitext (the use case for which we have built the system), we would want the Phoenix Urdu taggers to at least be able to tag the types of things that were successfully tagged by tsurgeon and lookup on the English side and then projected over word-aligned English and Urdu sentence pairs (our training data).

The 16 evaluations are described in Table 8.2.

For each evaluation, we measured the precision and recall based on exact matches (that is, the same word in the same sentence was tagged with the same tag label by both the candidate and the reference taggers). We also measured precision and recall based on a partial match, which we defined as the same word in the same sentence was tagged (with any tag label) by both the candidate and the reference taggers. Finally, we measured the precision and recall based on simply whether or not the two taggers tagged the same sentence at all. We call these three measures match exact, match any, and match sentence, and the results are presented for the Urdu evaluations, averaged over the four development sets, in Figure 8.7. The same results for the English evaluations, again averaged over the four development sets, are shown in Figure 8.8. These results indicate more sophisticated learned tagging is needed match the quality of rule-based taggers. Such taggers will likely need to incorporate more complex features such as sentence parse structure.

8.4 Human Evaluation of the Structure-based English Modality Modality Tagger

Post-workshop, one of the authors hand-evaluated the results of the rule-based modality tagger on a sample of the sentences.

Precision of the tagging was very good. Of the 116 English examples taken from SIMT LDC English training sentences, 95 tags, or 82%, were correct. Most errors were omissions from the modality lexicon or

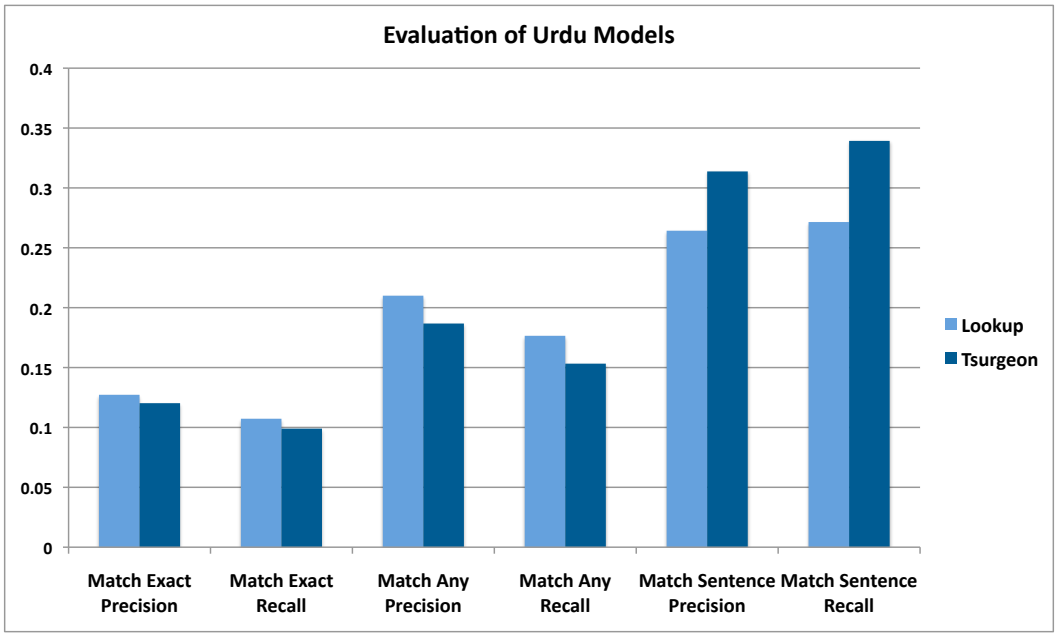


Figure 8.7: Evaluation on Urdu data, averaged over four development sets

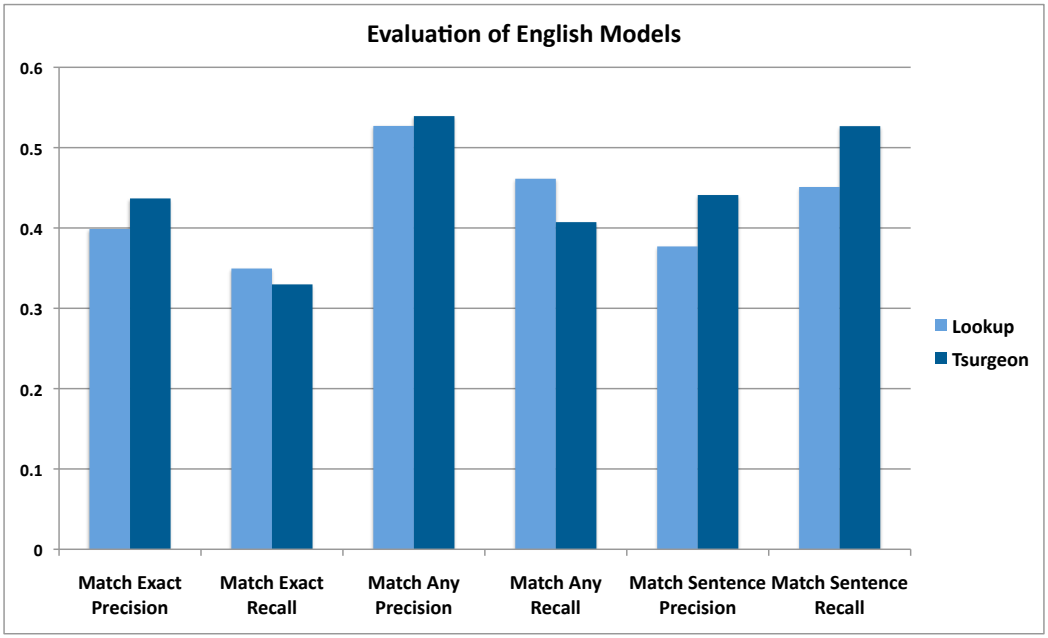


Figure 8.8: Evaluation on English data, averaged over four development sets

Language	Candidate	Reference
English	tsurgeon-Phoenix model applied to English dev.0	tsurgeon applied to English dev.0
English	tsurgeon-Phoenix model applied to English dev.1	tsurgeon applied to English dev.1
English	tsurgeon-Phoenix model applied to English dev.2	tsurgeon applied to English dev.2
English	tsurgeon-Phoenix model applied to English dev.3	tsurgeon applied to English dev.3
English	lookup-Phoenix model applied to English dev.0	lookup applied to English dev.0
English	lookup-Phoenix model applied to English dev.1	lookup applied to English dev.1
English	lookup-Phoenix model applied to English dev.2	lookup applied to English dev.2
English	lookup-Phoenix model applied to English dev.3	lookup applied to English dev.3
Urdu	tsurgeon-Phoenix model applied to Urdu dev.0	tsurgeon applied to English dev.0, projected to Urdu
Urdu	tsurgeon-Phoenix model applied to Urdu dev.1	tsurgeon applied to English dev.1, projected to Urdu
Urdu	tsurgeon-Phoenix model applied to Urdu dev.2	tsurgeon applied to English dev.2, projected to Urdu
Urdu	tsurgeon-Phoenix model applied to Urdu dev.3	tsurgeon applied to English dev.3, projected to Urdu
Urdu	lookup-Phoenix model applied to Urdu dev.0	lookup applied to English dev.0, projected to Urdu
Urdu	lookup-Phoenix model applied to Urdu dev.1	lookup applied to English dev.1, projected to Urdu
Urdu	lookup-Phoenix model applied to Urdu dev.2	lookup applied to English dev.2, projected to Urdu
Urdu	lookup-Phoenix model applied to Urdu dev.3	lookup applied to English dev.3, projected to Urdu

Table 8.2: Modality Evaluation Setup

special cases we have not yet addressed.

In terms of precision, sometimes the light verb or noun was the correct syntactic target, but not the correct semantic target.

- The decision *should* be *taken* on delayed cases on the basis of merit.

Sometimes, since the modality lexicon was used without respect to word sense, the wrong word sense was tagged. For example "attacked" was part of the lexicon with the intended sense of "succeed," as in "attacked the problem," but this did not often match the word sense for "attacked" in a newswire sentence.

- In Bayas, Sikhs *attacked a train* under cover of night and killed everyone.

A known limitation of the current rule-based approach is that it did not address more complex coordinate structures.

- Many large helicopters are *needed* to dispatch urgent relief materials to the many affected in far flung areas of the Neelam Valley and only America can help us in this regard.

With respect to recall, the tagger primarily missed special forms of negation.

- There was *no* place to seek shelter.
- The buildings should be reconstructed, *not* with RCC, but with the wood and steel sheets.

More complex constructional and phrasal triggers were also missed.

- President Pervaiz Musharraf has said that he will *not rest unless* the process of rehabilitation is completed.

Finally, we discovered some random lexical omissions (from our modality lexicon).

- It is not *possible* in the middle of winter to re-open the roads.

8.5 Evaluation of Modality Informing Machine Translation

As described in Section 3.4.1, the Cunei EBMT system was able to exploit the use of tags from our projected Urdu tagger on the source sentence (see Figure 3.3 and evaluation of Cunei results Figure 3.9).

We also used the grafting process described in Section 2.3 to allow modality inform the Joshua rule-based MT system. Training with modality tags grafted on the target English parse tree added 0.3 Bleu to Joshua score (see Section 2.15),

Examining the outputs of these translations, we find an example of how modalities can help inform the translation process. Without semantics, the translation misses the target of of the "try" modality:

- the cabinet meeting in said that ' we will try every possible to strengthen moderate elements in the palestinian authority .

By using modality semantics, the derived rules incorporate constraints that ensure the target of try (in this case "way") is included in the translation:

- the cabinet meeting in said that ' we will try every possible **way** to strengthen moderate elements in the palestinian authority .

Chapter 9

A HIVE-Aware Evaluation Measure

We set out to create a HIVE-Aware translation error rate metric, as none of the commonly used evaluation metrics – BLEU, METEOR, TER(p), etc – are particularly well suited for measuring performance of systems on HIVEs. BLEU in particular, almost certainly the most widely used and reported metric, considers HIVE tokens to be the same as all others and will assign high scores to hypothesis translations which differ dramatically in meaning from their corresponding references.

9.1 TER-based Metrics

Translation Error Rate (TER) (Snover et al., 2006) metrics operate by computing a word-based string edit distance between a hypothesis translation and one or more references. The more operations (insert, delete, substitute, etc) necessary to reconcile the two strings, the higher the reported Translation Error Rate. The simplest TER mechanism uses only four operations: match, when the words are identical; insert, when the hypothesis contains a word the reference does not; delete, for the reverse; and substitute, when the reference and hypothesis differ at a given position.

In an effort to improve the linguistic plausibility of TER alignments, metrics like TERp (Snover et al., 2009) have been created. TERp expands the vocabulary of string edit distance operations to include paraphrases (and a large set of consulted paraphrases derived from newswire text), WordNet synonymy, and Porter stem matches. It further allows bulk motion of parts of the hypothesis (at a cost), called shifts, in an effort to emulate linguistic alternatives.

The extant TERp codebase further offers a series of experimental features under the label of "word classes", which allow the vocabulary to be partitioned into (disjoint) sets. Costs for matching, shifting, insertion, and deletion may be set for each such set; costs for substitution, synonymy, and stem-wise matching may be set for each pair of such sets. A further experimental add-on allows entries in the "word classes" to be multiple words long.

9.2 HATERp Mechanisms

The HATERp effort consisted of the creation of preprocessor tools to feed more data to the HIVE-Aware TERp scoring metric, a series of (mostly small) changes to the functionality of the TERp codebase, and construction of word classes and corresponding weight tables. We discuss each of these in turn.

HATERp consumes modality- and named-entity-tagged parse trees of the reference sentences. The additional input is used in two ways: to POS-tag the references and to automatically expand the reference set by handling negated modal triggers. Our POS tag vocabulary is that of the WSJ annotation effort with all

name-like forms conflated to a single label; POS tags are used to guide special scoring for names and improve precision of modality trigger recall. Reference expansion attempts to make up for the nondisjointness of our modality annotations; in particular, some classes are negated forms of others. We therefore generate reference sentences with placeholders for the logically equivalent forms and allow HATERp to attempt alignments with these as well. We further expand any hyphenated named entities in the reference to offer both hyphenated and unhyphenated forms (with components as separate tokens).

The feature changes to TERp were few. We relaxed the restriction that word classes must be disjoint – instead the system now selects the cheapest of all available alternatives. Further, we augmented the edit distance function with a sub-word (*i.e.* character-wise) DICE similarity metric (though the choice of DICE was arbitrary; subsequent developers may replace the particular metric at will). This alignment option is currently restricted to words tagged by the preprocessor as names and gives us the ability to give partial credit – a smaller cost than the full substitution cost – for variant spellings of names.

We then created two sets of word classes. One set is composed of "functional" elements, including the closed classes of determiners and punctuation symbols, and as such is not terribly interesting; these classes exist mostly to constrain the set of alignments HATERp considers. The other set contains one class for each related family of modal triggers and forms the mechanism for evaluation of modality. The intra-class substitution costs are small, while the inter-class, insertion, and deletion costs are large: thereby, a system will not be heavily penalized for giving an equivalent alternative, but omission or spurious generation will be harshly scored.

9.3 Conclusion

All told, these mechanisms improve the linguistic plausibility of the alignments generated by the TERp scorer and offer tunable parameters specifically for scoring the forms of HIVE errors we wish to detect. All changes have been reported back to the TERp developers and with luck will be incorporated in future editions of TERp, enabling subsequent users to take advantage of our work.

Chapter 10

Future Directions

The overall goal of the SIMT SCALE was to develop several new ideas to use structured types in order to inform translation of high information value elements, or HIVEs, of foreign language text. The SCALE was very successful in achieving this goal. One of the most critical outcomes was the creation of extensible frameworks to enable the integration of linguistic structure into statistically trained machine translation systems. We focused on two facets of this integration: (1) The use of syntax in named-entity rule integration and modality tagging; (2) The use of semantics for MT and HIVE-Aware Evaluation.

Going forward, we intend to continue to test the impact of identifying HIVEs by explicitly integrating into SIMT other structured types, such as relations between entities. An example of the relationship Physical.Located between a person and a location is shown in figure 10.1 for the sentence *In the West Bank, a passenger was wounded.*

The Cunei software currently supports the output of structured annotation on English or other target language output, such that HIVEs are identified in the translation. The Joshua framework maintains this same annotation information internally. It would be a matter of minor adjustments to output HIVE annotation using Joshua. An important use case for producing structured output is to output just the translations of the desired HIVEs on demand, or to highlight HIVEs in translations for human analysts working directly with those translations. Another use of structured, automated output is to support applications which call for knowledge representation of natural language text.

SIMT results support the ingest of machine translation by automated analytics. Broadly speaking, entity extraction and normalization are a starting point for several applications, including document indexing on normalized HIVEs. An area of particular interest to the authors is automatic knowledge base population. English (realized in this work as either translation or transliteration) is a form of normalization for entities that are originally rendered in many different languages. HIVEs may be normalized and entered into a knowledge base. These HIVEs may be, for example, entities, facts about entities, or events. Modalities function as certainty values associated with events. Modalities merit their own entry into a knowledge base as a way to distinguish realized events from unrealized events, beliefs from certainties, etc, and importantly, to distinguish positive and negative instances of entities and events.

Many of the tasks which are described in this document provide exciting opportunities for follow-on work. One such task is to extend HIVE identification to the source language, in this instance, Urdu. One

Relation type	Arg1: PER	Arg2: LOC
Physical.Located	a **passenger**	the **West Bank**

Figure 10.1: Physical location relation for `In the West Bank, a passenger was wounded.`

could capitalize on entity or modality tagging in both English and Urdu during machine translation training and decoding, further constraining the translations. Tagging modality in any language is state-of-the-art work which requires further research and evaluation. The fact that identifying HIVEs of multiple types in the target language improved MT encourages incorporating similar methods on the source side.

The SIMT framework is available to answer the question of whether a fully developed semantics can be integrated with MT. Using Joshua, we can use ontological concepts as semantic tags grafted onto a parse tree during training in much the same way that we now annotate the trees with entity tags. Synchronous rules extracted for training would then be annotated with concepts. During the SCALE, an ontology was annotated with modality concepts as a precursor to such work.

Appendix A

Modality Lexicon

This section provides the modality lexicon, each entry of which consists of:

- A string of one or more words: for example, *should* or *have need of*. The words are inflected, so *require*, *required*, *requires*, and *requiring* are separate entries in the modality lexicon.
- A part of speech for each word: The part of speech helps us avoid irrelevant homophones such as the noun *can*.
- A modality: one of the fourteen modalities described above, further specified according to whether Negation is applicable. Specifically, the Modality classes included Require, Permit, NotPermit, Succeed, NotSucceed, Effort, NotEffort, Intend, Able, NotAble, Want, NotWant, FirmBelief, Belief, NotBelief, and Negation. Note: Not all words were implemented for each of these cases.
- A head word: the primary phrasal constituent to cover cases where an entry is a multi-word unit, e.g., the word *hope* in *hope for*.
- One or subcategorization codes: the codes are derived from syntactic codes provided in the *Longman's Dictionary of Contemporary English* (LDOCE). They include transitive verb, intransitive verb, transitive/intransitive verb plus infinitive verb phrase, etc. They are described in more detail below. The subcategorization codes indicate which syntactic templates are used by the structure-based tagger.

```
# Require
require''VB$Require &require, V3-passive-basic,
 V3-I3-basic, T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb,
 I5-CP-basic
required''VBD$Require &required, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, I5-CP-basic
required''VBN$Require &required, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, I5-CP-basic
requires''VBZ$Require &requires, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, I5-CP-basic
requiring''VBG$Require &requiring, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, I5-CP-basic
requirement''NN$Require &requirement, Noun-compound, NN-for-basic
required''JJ$Require &required, JJ-prenominal-basic
should''MD$Require &should, modal-auxiliary-basic
have''VB to''TO$Require &have, have-to
had''VBD to''TO$Require &had, have-to
```

```
had''VBN to''TO$Require &had, have-to
has''VBZ to''TO$Require &has, have-to
having''VB to''TO$Require &having, have-to
must''MD$Require &must, must-without-have
have''VB the''DT need''NN$Require &have, have-need-of
had''VBD the''DT need''NN$Require &had, have-need-of
had''VBN the''DT need''NN$Require &had, have-need-of
has''VBZ the''DT need''NN$Require &has, have-need-of
having''VBG the''DT need''NN$Require &having, have-need-of
in''IN need''NN of''IN$Require &need, in-need-of
need''VB$Require &need, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb,
 modal-auxiliary-basic
needed''VBD$Require &needed, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb,
 modal-auxiliary-basic
needed''VBN$Require &needed, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb,
 modal-auxiliary-basic
needs''VBZ$Require &needs, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb,
 modal-auxiliary-basic
needing''VBG$Require &needing, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb,
 modal-auxiliary-basic
needed''JJ$Require &need, JJ-prenominal-basic
need''NN$Require &need, NN-for-basic, NN-infinitive-basic
needs''NNS$Require &need, NN-for-basic, NN-infinitive-basic
obligate''VB$Require &obligate, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
obligated''VBD$Require &obligated, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
obligated''VBN$Require &obligated, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
obligates''VBZ$Require &obligates, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
obligating''VBG$Require &obligating, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
order''VB$Require &order, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, I5-CP-basic,
 X7-X9-basic
ordered''VBD$Require &ordered, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, I5-CP-basic,
 X7-X9-basic
ordered''VBN$Require &ordered, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, I5-CP-basic,
 X7-X9-basic
orders''VBZ$Require &orders, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, I5-CP-basic,
 X7-X9-basic
```

```
ordering''VBG$Require &ordering, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, I5-CP-basic,
 X7-X9-basic

# Permit
permit''VB$Permit &permit, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
permitted''VBD$Permit &permitted, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
permitted''VBN$Permit &permitted, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
permits''VBZ$Permit &permits, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
permitting''VBG$Permit &permitting, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
allow''VB$Permit &allow, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, X7-X9-basic
allowed''VBD$Permit &allowed, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, X7-X9-basic
allowed''VBN$Permit &allowed, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, X7-X9-basic
allows''VBZ$Permit &allows, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, X7-X9-basic
allowing''VBG$Permit &allowing, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-D1-verbs, T1-monotransitive-for-V3-verbs,
 T1-passive-for-V3-verb, D1-ditransitive-basic, X7-X9-basic
authorize''VB$Permit &authorize, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
authorized''VBD$Permit &authorized, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
authorized''VBN$Permit &authorized, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
authorizes''VBZ$Permit &authorizes, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
authorizing''VBG$Permit &authorizing, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb
enable''VB$Permit &enable, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
enabled''VBD$Permit &enabled, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
enabled''VBN$Permit &enabled, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
enables''VBZ$Permit &enables, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
enabling''VBG$Permit &enabling, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
endorse''VB$Permit &endorse, T1-monotransitive-basic,
 T1-passive-basic
endorsed''VBD$Permit &endorsed, T1-monotransitive-basic,
```

```
 T1-passive-basic
endorsed''VBN$Permit &endorsed, T1-monotransitive-basic,
 T1-passive-basic
endorses''VBZ$Permit &endorses, T1-monotransitive-basic,
 T1-passive-basic
endorsing''VBG$Permit &endorsing, T1-monotransitive-basic,
 T1-passive-basic
give''VB leave''NN$Permit &**NOT DONE**
gave''VBD leave''NN$Permit &**NOT DONE**
given''VBN leave''NN$Permit &**NOT DONE**
gives''VBZ permission''NN$Permit &**NOT DONE**
giving''VBG permission''NN$Permit &**NOT DONE**
give''VB leave''NN$Permit &**NOT DONE**
gave''VBD leave''NN$Permit &**NOT DONE**
given''VBN leave''NN$Permit &**NOT DONE**
gives''VBZ permission''NN$Permit &**NOT DONE**
giving''VBG permission''NN$Permit &**NOT DONE**
let''VB$Permit & let, let-s, let-vp
let''VBD$Permit & let, let-s, let-vp
let''VBN$Permit & let, let-s, let-vp
lets''VBZ$Permit & lets, let-s, let-vp
letting''VBG$Permit & letting, let-s, let-vp
okay''VB$Permit &okay, T1-monotransitive-basic, T1-passive-basic
okayed''VBD$Permit &okayed, T1-monotransitive-basic, T1-passive-basic
okayed''VBN$Permit &okayed, T1-monotransitive-basic, T1-passive-basic
okays''VBZ$Permit &okays, T1-monotransitive-basic, T1-passive-basic
okaying''VBG$Permit &okaying, T1-monotransitive-basic,
 T1-passive-basic
sanctify''VB$Permit &sanctify, T1-monotransitive-basic,
 T1-passive-basic
sanctified''VBD$Permit &sanctified, T1-monotransitive-basic,
 T1-passive-basic
sanctified''VBN$Permit &sanctified, T1-monotransitive-basic,
 T1-passive-basic
sanctifies''VBZ$Permit &sanctifies, T1-monotransitive-basic,
 T1-passive-basic
sanctifying''VBG$Permit &sanctifying, T1-monotransitive-basic,
 T1-passive-basic
sanction''VB$Permit &sanction, T1-monotransitive-basic,
 T1-passive-basic
sanctioned''VBD$Permit &sanctioned, T1-monotransitive-basic,
 T1-passive-basic
sanctioned''VBN$Permit &sanctioned, T1-monotransitive-basic,
 T1-passive-basic
sanctions''VBZ$Permit &sanctions, T1-monotransitive-basic,
 T1-passive-basic
sanctioning''VBG$Permit &sanctioning, T1-monotransitive-basic,
 T1-passive-basic
sign''VB off''RB$Permit &**NOT DONE**
signed''VBD off''RB$Permit &**NOT DONE**
signed''VBN off''RB$Permit &**NOT DONE**
signs''VBZ off''RB$Permit &**NOT DONE**
signing''VBG off''RB$Permit &**NOT DONE**
sign''VB on''RB$Permit &**NOT DONE**
```

```
signed''VBD on''RB$Permit &**NOT DONE**
signed''VBN on''RB$Permit &**NOT DONE**
signs''VBZ on''RB$Permit &**NOT DONE**
signing''VBG on''RB$Permit &**NOT DONE**
suffer''VB$Permit &suffer, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
suffered''VBD$Permit &suffered, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
suffered''VBN$Permit &suffered, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
suffers''VBZ$Permit &suffers, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
suffering''VBG$Permit &suffering, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
tolerate''VB$Permit &tolerate, T1-monotransitive-basic,
 T1-passive-basic
tolerated''VBD$Permit &tolerated, T1-monotransitive-basic,
 T1-passive-basic
tolerated''VBN$Permit &tolerated, T1-monotransitive-basic,
 T1-passive-basic
tolerates''VBZ$Permit &tolerates, T1-monotransitive-basic,
 T1-passive-basic
tolerating''VBG$Permit &tolerating, T1-monotransitive-basic,
 T1-passive-basic

# NotPermit
deny''VB$NotPermit &deny, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
denied''VBD$NotPermit &denied, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
denied''VBN$NotPermit &denied, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
denies''VBZ$NotPermit &denies, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
denying''VBG$NotPermit &denying, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
disallow''VB$NotPermit &disallow, T1-monotransitive-basic,
 T1-passive-basic
disallowed''VBD$NotPermit &disallowed, T1-monotransitive-basic,
 T1-passive-basic
disallowed''VBN$NotPermit &disallowed, T1-monotransitive-basic,
 T1-passive-basic
disallows''VBZ$NotPermit &disallows, T1-monotransitive-basic,
 T1-passive-basic
disallowing''VBG$NotPermit &disallowing, T1-monotransitive-basic,
 T1-passive-basic
refuse''VB$NotPermit &refuse, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, V3-passive-basic, V3-I3-basic
refused''VBD$NotPermit &refused, T1-monotransitive-for-D1-verbs,
```

```
 D1-ditransitive-basic, V3-passive-basic, V3-I3-basic
refused''VBN$NotPermit &refused, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, V3-passive-basic, V3-I3-basic
refuses''VBZ$NotPermit &refuses, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, V3-passive-basic, V3-I3-basic
refusing''VBG$NotPermit &refusing, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, V3-passive-basic, V3-I3-basic
exclude''VB$NotPermit &exclude, T1-monotransitive-basic,
 T1-passive-basic
excluded''VBD$NotPermit &excluded, T1-monotransitive-basic,
 T1-passive-basic
excluded''VBN$NotPermit &excluded, T1-monotransitive-basic,
 T1-passive-basic
excludes''VBZ$NotPermit &excludes, T1-monotransitive-basic,
 T1-passive-basic
excluding''VBG$NotPermit &excluding, T1-monotransitive-basic,
 T1-passive-basic
enjoin''VB from''IN$NotPermit &**NOT DONE**
enjoined''VBD from''IN$NotPermit &**NOT DONE**
enjoined''VBN from''IN$NotPermit &**NOT DONE**
enjoins''VBZ from''IN$NotPermit &**NOT DONE**
enjoining''VBG from''IN$NotPermit &**NOT DONE**
restrain''VB$NotPermit &restrain, T1-monotransitive-basic,
 T1-passive-basic
restrained''VBD$NotPermit &restrained, T1-monotransitive-basic,
 T1-passive-basic
restrained''VBN$NotPermit &restrained, T1-monotransitive-basic,
 T1-passive-basic
restrains''VBZ$NotPermit &restrains, T1-monotransitive-basic,
 T1-passive-basic
restraining''VBG$NotPermit &restraining, T1-monotransitive-basic,
 T1-passive-basic
veto''VB$NotPermit &veto, T1-monotransitive-basic, T1-passive-basic
vetoed''VBD$NotPermit &vetoed, T1-monotransitive-basic,
 T1-passive-basic
vetoed''VBN$NotPermit &vetoed, T1-monotransitive-basic,
 T1-passive-basic
vetoes''VBZ$NotPermit &vetoes, T1-monotransitive-basic,
 T1-passive-basic
vetoing''VBG$NotPermit &vetoing, T1-monotransitive-basic,
 T1-passive-basic

# Succeed
attain''VB$Succeed &attain, T1-monotransitive-basic, T1-passive-basic
attained''VBD$Succeed &attained, T1-monotransitive-basic,
 T1-passive-basic
attained''VBN$Succeed &attained, T1-monotransitive-basic,
 T1-passive-basic
attains''VBZ$Succeed &attains, T1-monotransitive-basic,
 T1-passive-basic
attaining''VBG$Succeed &attaining, T1-monotransitive-basic,
 T1-passive-basic
arrive''VB$Succeed &arrive, I-at-intransitive-basic.txt
arrived''VBD$Succeed &arrived, I-at-intransitive-basic.txt
```

```
arrived''VBN$Succeed &arrived, I-at-intransitive-basic.txt
arrives''VBZ$Succeed &arrives, I-at-intransitive-basic.txt
arriving''VBG$Succeed &arriving, I-at-intransitive-basic.txt
come''VB through''IN with''IN$Succeed &**NOT DONE**
came''VBD through''IN with''IN$Succeed &**NOT DONE**
came''VBN through''IN with''IN$Succeed &**NOT DONE**
comes''VBZ through''IN with''IN$Succeed &**NOT DONE**
coming''VBG through''IN with''IN$Succeed &**NOT DONE**
manage''VB$Succeed &manage, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
managed''VBD$Succeed &managed, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
managed''VBN$Succeed &managed, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
manages''VBZ$Succeed &manages, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
managing''VBG$Succeed &managing, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
manage''VB to''TO$Succeed &**NOT DONE**
managed''VBD to''TO$Succeed &**NOT DONE**
managed''VBN to''TO$Succeed &**NOT DONE**
manages''VBZ to''TO$Succeed &**NOT DONE**
managing''VBG to''TO$Succeed &**NOT DONE**
reach''VB$Succeed &reach, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
reached''VBD$Succeed &reached, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
reached''VBN$Succeed &reached, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
reaches''VBZ$Succeed &reaches, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
reaching''VBG$Succeed &reaching, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
succeed''VB$Succeed &succeed, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
succeeded''VBD$Succeed &succeeded, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
succeeded''VBN$Succeed &succeeded, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
succeeds''VBZ$Succeed &succeeds, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
succeeding''VBG$Succeed &succeeding, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
successful''JJ$Succeed & successful, JJ-prenominal-basic,
 JJ-predicative-for-JJ-in, JJ-in-basic, JJ-in-VBG
accomplish''VB$Succeed &accomplish, T1-monotransitive-basic,
 T1-passive-basic
accomplished''VBD$Succeed &accomplished, T1-monotransitive-basic,
 T1-passive-basic
accomplished''VBN$Succeed &accomplished, T1-monotransitive-basic,
 T1-passive-basic
accomplishes''VBZ$Succeed &accomplishes, T1-monotransitive-basic,
 T1-passive-basic
accomplishing''VBG$Succeed &accomplishing, T1-monotransitive-basic,
```

```
 T1-passive-basic
achieve''VB$Succeed &achieve, T1-monotransitive-basic,
 T1-passive-basic
achieved''VBD$Succeed &achieved, T1-monotransitive-basic,
 T1-passive-basic
achieved''VBN$Succeed &achieved, T1-monotransitive-basic,
 T1-passive-basic
achieves''VBZ$Succeed &achieves, T1-monotransitive-basic,
 T1-passive-basic
achieving''VBG$Succeed &achieving, T1-monotransitive-basic,
 T1-passive-basic
acquire''VB$Succeed &acquire, T1-monotransitive-basic,
 T1-passive-basic
acquired''VBD$Succeed &acquired, T1-monotransitive-basic,
 T1-passive-basic
acquired''VBN$Succeed &acquired, T1-monotransitive-basic,
 T1-passive-basic
acquires''VBZ$Succeed &acquires, T1-monotransitive-basic,
 T1-passive-basic
acquiring''VBG$Succeed &acquiring, T1-monotransitive-basic,
 T1-passive-basic
fulfill''VB$Succeed &fulfill, T1-monotransitive-basic,
 T1-passive-basic
fulfilled''VBD$Succeed &fulfilled, T1-monotransitive-basic,
 T1-passive-basic
fulfilled''VBN$Succeed &fulfilled, T1-monotransitive-basic,
 T1-passive-basic
fulfills''VBZ$Succeed &fulfills, T1-monotransitive-basic,
 T1-passive-basic
fulfilling''VBG$Succeed &fulfilling, T1-monotransitive-basic,
 T1-passive-basic
obtain''VB$Succeed &obtain, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
obtained''VBD$Succeed &obtained, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
obtained''VBN$Succeed &obtained, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
obtains''VBZ$Succeed &obtains, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
obtaining''VBG$Succeed &obtaining, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, L9-basic
prevail''VB$Succeed &prevail, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
prevailed''VBD$Succeed &prevailed, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
prevailed''VBN$Succeed &prevailed, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
prevails''VBZ$Succeed &prevails, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
prevailing''VBG$Succeed &prevailing, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
pull''VB off''RB$Succeed &**NOT DONE**
pulled''VBD off''RB$Succeed &**NOT DONE**
pulled''VBN off''RB$Succeed &**NOT DONE**
```

```
pulls''VBZ off''RB$Succeed &**NOT DONE**
pulling''VBG off''RB$Succeed &**NOT DONE**
win''VB$Succeed &win, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
won''VBD$Succeed &won, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
won''VBN$Succeed &won, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
wins''VBZ$Succeed &wins, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
winning''VBG$Succeed &winning, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic


# NotSucceed
fail''VB$NotSucceed &fail, V3-passive-basic, V3-I3-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 T1-monotransitive-basic, T1-passive-basic, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
failed''VBD$NotSucceed &failed, V3-passive-basic, V3-I3-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 T1-monotransitive-basic, T1-passive-basic, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
failed''VBN$NotSucceed &failed, V3-passive-basic, V3-I3-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 T1-monotransitive-basic, T1-passive-basic, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
fails''VBZ$NotSucceed &fails, V3-passive-basic, V3-I3-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 T1-monotransitive-basic, T1-passive-basic, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
failing''VBG$NotSucceed &failing, V3-passive-basic, V3-I3-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 T1-monotransitive-basic, T1-passive-basic, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
abort''VB$NotSucceed &abort, T1-monotransitive-basic,
 T1-passive-basic
aborted''VBD$NotSucceed &aborted, T1-monotransitive-basic,
 T1-passive-basic
aborted''VBN$NotSucceed &aborted, T1-monotransitive-basic,
 T1-passive-basic
aborts''VBZ$NotSucceed &aborts, T1-monotransitive-basic,
 T1-passive-basic
aborting''VBG$NotSucceed &aborting, T1-monotransitive-basic,
 T1-passive-basic
fall''VB short''JJ$NotSucceed & fall, fall-short
fell''VBD short''JJ$NotSucceed & fell, fall-short
fallen''VBN short''JJ$NotSucceed & fallen, fall-short
falls''VBZ short''JJ$NotSucceed & falls, fall-short
falling''VBG short''JJ$NotSucceed & falling, fall-short
fizzle''VB$NotSucceed &fizzle, I-intransitive-basic
fizzled''VBD$NotSucceed &fizzled, I-intransitive-basic
fizzled''VBN$NotSucceed &fizzled, I-intransitive-basic
fizzles''VBZ$NotSucceed &fizzles, I-intransitive-basic
```

```
fizzling''VBG$NotSucceed &fizzling, I-intransitive-basic
flop''VB$NotSucceed &flop, I-intransitive-basic, L9-basic
flopped''VBD$NotSucceed &flopped, I-intransitive-basic, L9-basic
flopped''VBN$NotSucceed &flopped, I-intransitive-basic, L9-basic
flops''VBZ$NotSucceed &flops, I-intransitive-basic, L9-basic
flopping''VBG$NotSucceed &flopping, I-intransitive-basic, L9-basic
short''JJ of''IN$NotSucceed & short, short-of, short-of-vbg
unsuccessful''JJ$NotSucceed & unsuccessful, JJ-prenominal-basic,
 JJ-predicative-for-JJ-in, JJ-in-basic, JJ-in-VBG
useless''JJ$NotSucceed & useless, JJ-in-VBG, JJ-as-basic

# Effort
try''VB$Effort &try, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
tried''VBD$Effort &tried, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
tried''VBN$Effort &tried, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
tries''VBZ$Effort &tries, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
trying''VBG$Effort &trying, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
aim''VB$Effort &aim, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aimed''VBD$Effort &aimed, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aimed''VBN$Effort &aimed, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aims''VBZ$Effort &aims, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aiming''VBG$Effort &aiming, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
aspire''VB$Effort &aspire, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspired''VBD$Effort &aspires, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspired''VBN$Effort &aspired, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspires''VBZ$Effort &aspires, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspiring''VBG$Effort &aspiring, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
attack''VB$Effort &attack, T1-monotransitive-basic, T1-passive-basic
attacked''VBD$Effort &attacked, T1-monotransitive-basic,
 T1-passive-basic
attacked''VBN$Effort &attacked, T1-monotransitive-basic,
 T1-passive-basic
attacks''VBZ$Effort &attacks, T1-monotransitive-basic,
 T1-passive-basic
attacking''VBG$Effort &attacking, T1-monotransitive-basic,
 T1-passive-basic
attempt''VB$Effort &attempt, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
attempted''VBD$Effort &attempted, T1-monotransitive-basic,
```

```
 T1-passive-basic, V3-passive-basic, V3-I3-basic
attempted''VBN$Effort &attempted, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
attempts''VBZ$Effort &attempts, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
attempting''VBG$Effort &attempting, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
contend''VB$Effort &contend, I5-CP-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
contended''VBD$Effort &contended, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG,
 I-with-intransitive-basic, I-with-intransitive-VBG
contended''VBN$Effort &contended, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG,
 I-with-intransitive-basic, I-with-intransitive-VBG
contends''VBZ$Effort &contends, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG,
 I-with-intransitive-basic, I-with-intransitive-VBG
contending''VBG$Effort &contending, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG,
 I-with-intransitive-basic, I-with-intransitive-VBG
endeavor''VB$Effort &**NOT DONE**
endeavored''VBD$Effort &**NOT DONE**
endeavored''VBN$Effort &**NOT DONE**
endeavors''VBZ$Effort &**NOT DONE**
endeavoring''VBG$Effort &**NOT DONE**
seek''VB$Effort &seek, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
sought''VBD$Effort &sought, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
sought''VBN$Effort &sought, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
seeks''VBZ$Effort &seeks, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
seeking''VBG$Effort &seeking, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
strive''VB$Effort &strive, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
strove''VBD$Effort &strove, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
strove''VBN$Effort &strove, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
strives''VBZ$Effort &strives, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
striving''VBG$Effort &striving, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
tackle''VB$Effort &tackle, T1-monotransitive-basic, T1-passive-basic
tackled''VBD$Effort &tackled, T1-monotransitive-basic,
```

```
 T1-passive-basic
tackled''VBN$Effort &tackled, T1-monotransitive-basic,
 T1-passive-basic
tackles''VBZ$Effort &tackles, T1-monotransitive-basic,
 T1-passive-basic
tackling''VBG$Effort &tackling, T1-monotransitive-basic,
 T1-passive-basic
undertake''VB$Effort &undertake, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
undertook''VBD$Effort &undertook, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
undertook''VBN$Effort &undertook, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
undertakes''VBZ$Effort &undertakes, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
undertaking''VBG$Effort &undertaking, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
venture''VB$Effort &venture, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
ventured''VBD$Effort &ventured, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
ventured''VBN$Effort &ventured, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
ventures''VBZ$Effort &ventures, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
venturing''VBG$Effort &venturing, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
work''VB$Effort &work, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I-at-intransitive-basic.txt,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
worked''VBD$Effort &worked, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I-at-intransitive-basic.txt,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
worked''VBN$Effort &worked, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I-at-intransitive-basic.txt,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
works''VBZ$Effort &works, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I-at-intransitive-basic.txt,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
working''VBG$Effort &working, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I-at-intransitive-basic.txt,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
wrangle''VB$Effort &wrangle, I-with-intransitive-basic,
 I-with-intransitive-VBG
wrangled''VBD$Effort &wrangled, I-with-intransitive-basic,
 I-with-intransitive-VBG
wrangled''VBN$Effort &wrangled, I-with-intransitive-basic,
 I-with-intransitive-VBG
wrangles''VBZ$Effort &wrangles, I-with-intransitive-basic,
 I-with-intransitive-VBG
wrangling''VBG$Effort &wrangling, I-with-intransitive-basic,
 I-with-intransitive-VBG
engineer''VB$Effort &engineer, T1-monotransitive-basic,
 T1-passive-basic
```

```
engineered''VBD$Effort &engineered, T1-monotransitive-basic,
 T1-passive-basic
engineered''VBN$Effort &engineered, T1-monotransitive-basic,
 T1-passive-basic
engineers''VBZ$Effort &engineers, T1-monotransitive-basic,
 T1-passive-basic
engineering''VBG$Effort &engineering, T1-monotransitive-basic,
 T1-passive-basic
organize''VB$Effort &organize, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
organized''VBD$Effort &organized, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
organized''VBN$Effort &organized, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
organizes''VBZ$Effort &organizes, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
organizing''VBG$Effort &organizing, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic

# NotEffort
abstain''VB$NotEffort &abstain, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
abstained''VBD$NotEffort &abstained, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
abstained''VBN$NotEffort &abstained, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
abstains''VBZ$NotEffort &abstains, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
abstaining''VBG$NotEffort &abstaining, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
avoid''VB$NotEffort &avoid, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
avoided''VBD$NotEffort &avoided, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
avoided''VBN$NotEffort &avoided, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
avoids''VBZ$NotEffort &avoids, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
avoiding''VBG$NotEffort &avoiding, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
decline''VB$NotEffort &decline, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
declined''VBD$NotEffort &declined, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
declined''VBN$NotEffort &declined, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
declines''VBZ$NotEffort &declines, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
declining''VBG$NotEffort &declining, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
forgo''VB$NotEffort &forgo, T1-monotransitive-basic, T1-passive-basic
forgone''VBD$NotEffort &forgone, T1-monotransitive-basic,
 T1-passive-basic
forgone''VBN$NotEffort &forgone, T1-monotransitive-basic,
```

```
 T1-passive-basic
forgoes''VBZ$NotEffort &forgoes, T1-monotransitive-basic,
 T1-passive-basic
forgoing''VBG$NotEffort &forgoing, T1-monotransitive-basic,
 T1-passive-basic
pass''VB$NotEffort &pass, L9-basic, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, X7-X9-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
passed''VBD$NotEffort &passed, L9-basic,
 T1-monotransitive-for-D1-verbs, D1-ditransitive-basic, X7-X9-basic,
 V3-I3-basic, I-Prep-intransitive-basic, I-Prep-intransitive-VBG
passed''VBN$NotEffort &passed, L9-basic,
 T1-monotransitive-for-D1-verbs, D1-ditransitive-basic, X7-X9-basic,
 V3-I3-basic, I-Prep-intransitive-basic, I-Prep-intransitive-VBG
passes''VBZ$NotEffort &passes, L9-basic,
 T1-monotransitive-for-D1-verbs, D1-ditransitive-basic, X7-X9-basic,
 V3-I3-basic, I-Prep-intransitive-basic, I-Prep-intransitive-VBG
passing''VBG$NotEffort &passing, L9-basic,
 T1-monotransitive-for-D1-verbs, D1-ditransitive-basic, X7-X9-basic,
 V3-I3-basic, I-Prep-intransitive-basic, I-Prep-intransitive-VBG


# Intend
intend''VB$Intend &intend, V3-passive-basic, V3-I3-basic,
 X7-X9-basic, I5-CP-basic
intended''VBD$Intend &intended, V3-passive-basic, V3-I3-basic,
 X7-X9-basic, I5-CP-basic
intended''VBN$Intend &intended, V3-passive-basic, V3-I3-basic,
 X7-X9-basic, I5-CP-basic
intends''VBZ$Intend &intends, V3-passive-basic, V3-I3-basic,
 X7-X9-basic, I5-CP-basic
intending''VBG$Intend &intending, V3-passive-basic, V3-I3-basic,
 X7-X9-basic, I5-CP-basic
plan''VB$Intend &plan, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
planned''VBD$Intend &planned, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
planned''VBN$Intend &planned, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
plans''VBZ$Intend &plans, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic
planning''VBG$Intend &planning, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
aim''VB$Intend &aim, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aimed''VBD$Intend &aimed, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aimed''VBN$Intend &aimed, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aims''VBZ$Intend &aims, T1-monotransitive-basic, T1-passive-basic,
 V3-I3-basic
aiming''VBG$Intend &aiming, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
contemplate''VB$Intend &contemplate, T1-monotransitive-basic,
```

```
 T1-passive-basic, V3-I3-basic, I5-CP-basic
contemplated''VBD$Intend &contemplated, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, I5-CP-basic
contemplated''VBN$Intend &contemplated, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, I5-CP-basic
contemplates''VBZ$Intend &contemplates, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, I5-CP-basic
contemplating''VBG$Intend &contemplating, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, I5-CP-basic
design''VB$Intend &design, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
designed''VBD$Intend &designed, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
designed''VBN$Intend &designed, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
designs''VBZ$Intend &designs, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
designing''VBG$Intend &designing, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
hope''VB to''TO$Intend &**NOT DONE**
hoped''VBD to''TO$Intend &**NOT DONE**
hoped''VBN to''TO$Intend &**NOT DONE**
hopes''VBZ to''TO$Intend &**NOT DONE**
hoping''VBG to''TO$Intend &**NOT DONE**
indicate''VB$Intend &indicate, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
indicated''VBD$Intend &indicated, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
indicated''VBN$Intend &indicated, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
indicates''VBZ$Intend &indicates, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
indicating''VBG$Intend &indicating, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
ordain''VB$Intend &ordain, T1-monotransitive-basic, T1-passive-basic,
 I5-CP-basic
ordained''VBD$Intend &ordained, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
ordained''VBN$Intend &ordained, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
ordains''VBZ$Intend &ordains, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
ordaining''VBG$Intend &ordaining, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
plot''VB$Intend &plot, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic, I5-CP-basic
plotted''VBD$Intend &plotted, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
plotted''VBN$Intend &plotted, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
plots''VBZ$Intend &plots, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic, I5-CP-basic
plotting''VBG$Intend &plotting, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
```

```
propose''VB$Intend &propose, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
proposed''VBD$Intend &proposed, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
proposed''VBN$Intend &proposed, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
proposes''VBZ$Intend &proposes, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
proposing''VBG$Intend &proposing, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
resolve''VB$Intend &resolve, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
resolved''VBD$Intend &resolved, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
resolved''VBN$Intend &resolved, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
resolves''VBZ$Intend &resolves, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
resolving''VBG$Intend &resolving, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
scheme''VB$Intend &scheme, V3-I3-basic
schemed''VBD$Intend &schemed, V3-I3-basic
schemed''VBN$Intend &schemed, V3-I3-basic
schemes''VBZ$Intend &schemes, V3-I3-basic
scheming''VBG$Intend &scheming, V3-I3-basic
signify''VB$Intend &signify, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
signified''VBD$Intend &signified, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
signified''VBN$Intend &signified, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
signifies''VBZ$Intend &signifies, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
signifying''VBG$Intend &signifying, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
strive''VB$Intend &strive, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
strived''VBD$Intend &strived, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
strived''VBN$Intend &strived, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
strives''VBZ$Intend &strives, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
striving''VBG$Intend &striving, L9-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG, I-with-intransitive-basic,
 I-with-intransitive-VBG
concoct''VB$Intend &concoct, T1-monotransitive-basic,
 T1-passive-basic
concocted''VBD$Intend &concocted, T1-monotransitive-basic,
 T1-passive-basic
```

```
concocted''VBN$Intend &concocted, T1-monotransitive-basic,
 T1-passive-basic
concocts''VBZ$Intend &concocts, T1-monotransitive-basic,
 T1-passive-basic
concocting''VBG$Intend &concocting, T1-monotransitive-basic,
 T1-passive-basic
conspire''VB$Intend &conspire, V3-I3-basic
conspired''VBD$Intend &conspired, V3-I3-basic
conspired''VBN$Intend &conspired, V3-I3-basic
conspires''VBZ$Intend &conspires, V3-I3-basic
conspiring''VBG$Intend &conspiring, V3-I3-basic
contrive''VB$Intend &contrive, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
contrived''VBD$Intend &contrived, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
contrived''VBN$Intend &contrived, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
contrives''VBZ$Intend &contrives, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
contriving''VBG$Intend &contriving, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic
devise''VB$Intend &devise, T1-monotransitive-basic, T1-passive-basic
devised''VBD$Intend &devised, T1-monotransitive-basic,
 T1-passive-basic
devised''VBN$Intend &devised, T1-monotransitive-basic,
 T1-passive-basic
devises''VBZ$Intend &devises, T1-monotransitive-basic,
 T1-passive-basic
devising''VBG$Intend &devising, T1-monotransitive-basic,
 T1-passive-basic

# Able
capable''JJ of''IN$Able& capable, JJ-of-basic, JJ-of-VBG
able''JJ to''TO$Able & able, JJ-infinitive
can''MD$Able & can, modal-auxiliary-basic
could''MD$Able & could, modal-auxiliary-basic
ready''JJ$Able & ready, JJ-infinitive

# NotAble
powerless''JJ$NotAble & powerless, JJ-infinitive
unable''JJ$NotAble & unable, JJ-infinitive

# Want
want''VB$Want &want, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
wanted''VBD$Want &wanted, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
wanted''VBN$Want &wanted, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
wants''VBZ$Want &wants, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
wanting''VBG$Want &wanting, V3-passive-basic, V3-I3-basic,
 T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb, X7-X9-basic
hope''VB$Want &hope, V3-passive-basic, V3-I3-basic, I5-CP-basic
```

```
hopes''VBD$Want &hopes, V3-passive-basic, V3-I3-basic, I5-CP-basic
hopes''VBN$Want &hopes, V3-passive-basic, V3-I3-basic, I5-CP-basic
hoped''VBZ$Want &hoped, V3-passive-basic, V3-I3-basic, I5-CP-basic
hoping''VBG$Want &hoping, V3-passive-basic, V3-I3-basic, I5-CP-basic
wish''VB$Want &wish, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic, I5-CP-basic
wishes''VBD$Want &wishes, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic, I5-CP-basic
wishes''VBN$Want &wishes, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic, I5-CP-basic
wished''VBZ$Want &wished, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic, I5-CP-basic
wishing''VBG$Want &wishing, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic, I5-CP-basic
request''VB$Want &request, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
requests''VBD$Want &requests, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
requests''VBN$Want &requests, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
requested''VBZ$Want &requested, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
requesting''VBG$Want &requesting, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
aspire''VB$Want &aspire, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspires''VBD$Want &aspires, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspires''VBN$Want &aspires, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspired''VBZ$Want &aspired, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
aspiring''VBG$Want &aspiring, V3-passive-basic, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG
covet''VB$Want &covet, T1-monotransitive-basic, T1-passive-basic
covets''VBD$Want &covets, T1-monotransitive-basic, T1-passive-basic
covets''VBN$Want &covets, T1-monotransitive-basic, T1-passive-basic
coveted''VBZ$Want &coveted, T1-monotransitive-basic, T1-passive-basic
coveting''VBG$Want &coveting, T1-monotransitive-basic,
 T1-passive-basic
crave''VB$Want &crave, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
craves''VBD$Want &craves, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
craves''VBN$Want &craves, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
craved''VBZ$Want &craved, L9-basic, T1-monotransitive-basic,
 T1-passive-basic
craving''VBG$Want &craving, L9-basic, T1-monotransitive-basic,
```

```
 T1-passive-basic
fancy''VB$Want &fancy, V3-I3-basic, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
fancied''VBD$Want &fancied, V3-I3-basic, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
fancied''VBN$Want &fancied, V3-I3-basic, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
fancies''VBZ$Want &fancies, V3-I3-basic, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
fancying''VBG$Want &fancying, V3-I3-basic, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
hunger''VB$Want &hunger, I-for-intransitive-basic,
 I-for-intransitive-VBG
hungered''VBD$Want &hungered, I-for-intransitive-basic,
 I-for-intransitive-VBG
hungered''VBN$Want &hungered, I-for-intransitive-basic,
 I-for-intransitive-VBG
hungers''VBZ$Want &hungers, I-for-intransitive-basic,
 I-for-intransitive-VBG
hungering''VBG$Want &hungering, I-for-intransitive-basic,
 I-for-intransitive-VBG
incline''VB toward''IN$Want &**NOT DONE**
inclined''VBD toward''IN$Want &**NOT DONE**
inclined''VBN toward''IN$Want &**NOT DONE**
inclines''VBZ toward''IN$Want &**NOT DONE**
inclining''VBG toward''IN$Want &**NOT DONE**
incline''VB towards''IN$Want &**NOT DONE**
inclined''VBD towards''IN$Want &**NOT DONE**
inclined''VBN towards''IN$Want &**NOT DONE**
inclines''VBZ towards''IN$Want &**NOT DONE**
inclining''VBG towards''IN$Want &**NOT DONE**
incline''VB to''TO$Want &**NOT DONE**
inclined''VBD to''TO$Want &**NOT DONE**
inclined''VBN to''TO$Want &**NOT DONE**
inclines''VBZ to''TO$Want &**NOT DONE**
inclining''VBG to''TO$Want &**NOT DONE**
long''VB$Want &long, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
longed''VBD$Want &longed, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
longed''VBN$Want &longed, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
longs''VBZ$Want &longs, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
longing''VBG$Want &longing, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
lust''VB$Want &lust, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
lusted''VBD$Want &lusted, 3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
lusted''VBN$Want &lusted, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
lusts''VBZ$Want &lusts, 3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
```

```
lusting''VBG$Want &lusting, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
need''VB$Want &need, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, modal-auxiliary-basic
needed''VBD$Want &needed, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic,
 modal-auxiliary-basic
needed''VBN$Want &needed, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic,
 modal-auxiliary-basic
needs''VBZ$Want &needs, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, modal-auxiliary-basic
needing''VBG$Want &needing, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic,
 modal-auxiliary-basic
pine''VB$Want &pine, L9-basic, V3-I3-basic, I-for-intransitive-basic,
 I-for-intransitive-VBG
pined''VBD$Want &pined, L9-basic, V3-I3-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
pined''VBN$Want &pined, L9-basic, V3-I3-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
pines''VBZ$Want &pines, L9-basic, V3-I3-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
pining''VBG$Want &pining, L9-basic, V3-I3-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
prefer''VB$Want &prefer, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
preferred''VBD$Want &preferred, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
preferred''VBN$Want &preferred, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
prefers''VBZ$Want &prefers, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
preferring''VBG$Want &preferring, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
beg''VB$Want &beg, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
begged''VBD$Want &begged, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
begged''VBN$Want &begged, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
begs''VBZ$Want &begs, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG, I5-CP-basic,
```

```
 I-for-intransitive-basic, I-for-intransitive-VBG
begging''VBG$Want &begging, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic,
 I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
 I-Prep-intransitive-basic, I-Prep-intransitive-VBG, I5-CP-basic,
 I-for-intransitive-basic, I-for-intransitive-VBG
beseech''VB$Want &beseech, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
beseeched''VBD$Want &beseeched, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
beseeched''VBN$Want &beseeched, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
beseeches''VBZ$Want &beseeches, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
beseeching''VBG$Want &beseeching, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
entreat''VB$Want &entreat, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
entreated''VBD$Want &entreated, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
entreated''VBN$Want &entreated, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
entreats''VBZ$Want &entreats, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
entreating''VBG$Want &entreating, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
petition''VB$Want &petition, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
petitioned''VBD$Want &petitioned, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
petitioned''VBN$Want &petitioned, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
petitions''VBZ$Want &petitions, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
petitioning''VBG$Want &petitioning, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
promote''VB$Want &promote, T1-monotransitive-basic, T1-passive-basic
promoted''VBD$Want &promoted, T1-monotransitive-basic,
 T1-passive-basic
promoted''VBN$Want &promoted, T1-monotransitive-basic,
 T1-passive-basic
promotes''VBZ$Want &promotes, T1-monotransitive-basic,
 T1-passive-basic
promoting''VBG$Want &promoting, T1-monotransitive-basic,
 T1-passive-basic
solicit''VB$Want &solicit, T1-monotransitive-basic, T1-passive-basic
solicited''VBD$Want &solicited, T1-monotransitive-basic,
 T1-passive-basic
solicited''VBN$Want &solicited, T1-monotransitive-basic,
 T1-passive-basic
solicits''VBZ$Want &solicits, T1-monotransitive-basic,
 T1-passive-basic
soliciting''VBG$Want &soliciting, T1-monotransitive-basic,
 T1-passive-basic
```

```
# NotWant
despise''VB$NotWant &despise, T1-monotransitive-basic,
 T1-passive-basic
despised''VBD$NotWant &despised, T1-monotransitive-basic,
 T1-passive-basic
despised''VBN$NotWant &despised, T1-monotransitive-basic,
 T1-passive-basic
despises''VBZ$NotWant &despises, T1-monotransitive-basic,
 T1-passive-basic
despising''VBG$NotWant &despising, T1-monotransitive-basic,
 T1-passive-basic
dislike''VB$NotWant &dislike, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
disliked''VBD$NotWant &disliked, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
disliked''VBN$NotWant &disliked, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
dislikes''VBZ$NotWant &dislikes, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
disliking''VBG$NotWant &disliking, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic
hate''VB$NotWant &hate, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic
hated''VBD$NotWant &hated, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
hated''VBN$NotWant &hated, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
hates''VBZ$NotWant &hates, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
hating''VBG$NotWant &hating, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
despair''VB$NotWant &despair, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
despaired''VBD$NotWant &despaired, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
despaired''VBN$NotWant &despaired, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
despairs''VBZ$NotWant &despairs, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
despairing''VBG$NotWant &despairing, I-Prep-intransitive-basic,
 I-Prep-intransitive-VBG
fear''VB$NotWant &fear, T1-monotransitive-basic, T1-passive-basic
feared''VBD$NotWant &feared, T1-monotransitive-basic,
 T1-passive-basic
feared''VBN$NotWant &feared, T1-monotransitive-basic,
 T1-passive-basic
fears''VBZ$NotWant &fears, T1-monotransitive-basic, T1-passive-basic
fearing''VBG$NotWant &fearing, T1-monotransitive-basic,
 T1-passive-basic

# FirmBelief
certain''JJ$FirmBelief &**NOT DONE**
```

```
convinced''VB$FirmBelief &convinced, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
convinced''VBD$FirmBelief &convinced, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
convinced''VBN$FirmBelief &convinced, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
convinces''VBZ$FirmBelief &convinces, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
convincing''VBG$FirmBelief &convincing,
 T1-monotransitive-for-V3-verbs, V3-I3-basic, T1-passive-for-V3-verb,
 V3-passive-basic
know''VB$FirmBelief &know, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
knew''VBD$FirmBelief &knew, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
knew''VBN$FirmBelief &knew, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
knows''VBZ$FirmBelief &knows, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
knowing''VBG$FirmBelief &knowing, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
sure''JJ$FirmBelief &**NOT DONE**
destined''JJ$FirmBelief &**NOT DONE**
determined''VB$FirmBelief &**NOT DONE**
infallible''JJ$FirmBelief &**NOT DONE**
irrefutable''JJ$FirmBelief &**NOT DONE**
unambiguous''JJ$FirmBelief &**NOT DONE**
undeniable''JJ$FirmBelief &**NOT DONE**
unerring''JJ$FirmBelief &**NOT DONE**
unmistakable''JJ$FirmBelief &**NOT DONE**
verifiable''JJ$FirmBelief &**NOT DONE**
assure''VB$FirmBelief &assure, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
assured''VBD$FirmBelief &assured, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
assured''VBN$FirmBelief &assured, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
assures''VBZ$FirmBelief &assures, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
assuring''VBG$FirmBelief &assuring, T1-monotransitive-for-D1-verbs,
 D1-ditransitive-basic
satisfy''VB$FirmBelief &satisfy, T1-monotransitive-basic,
 T1-passive-basic
satisfied''VBD$FirmBelief &satisfied, T1-monotransitive-basic,
 T1-passive-basic
satisfied''VBN$FirmBelief &satisfied, T1-monotransitive-basic,
 T1-passive-basic
satisfies''VBZ$FirmBelief &satisfies, T1-monotransitive-basic,
 T1-passive-basic
satisfying''VBG$FirmBelief &satisfying, T1-monotransitive-basic,
 T1-passive-basic
take''VB as''NN gospel''IN$FirmBelief &**NOT DONE**
took''VBD as''NN gospel''IN$FirmBelief &**NOT DONE**
taken''VBN as''NN gospel''IN$FirmBelief &**NOT DONE**
```

```
takes''VBZ as''NN gospel''IN$FirmBelief &**NOT DONE**
taking''VBG as''NN gospel''IN$FirmBelief &**NOT DONE**

# Belief
appear''VB$Belief &appear, L9-basic, V3-I3-basic
appeared''VBD$Belief &appeared, L9-basic, V3-I3-basic
appeared''VBN$Belief &appeared, L9-basic, V3-I3-basic
appears''VBZ$Belief &appears, L9-basic, V3-I3-basic
appearing''VBG$Belief &appearing, L9-basic, V3-I3-basic
believe''VB$Belief &believe, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
believed''VBD$Belief &believed, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
believed''VBN$Belief &believed, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
believes''VBZ$Belief &believes, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
believing''VBG$Belief &believing, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
expect''VB$Belief &expect, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
expected''VBD$Belief &expected, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
expected''VBN$Belief &expected, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
expects''VBZ$Belief &expects, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
expecting''VBG$Belief &expecting, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
likely''JJ$Belief &**NOT DONE**
maybe''RB$Belief &**NOT DONE**
may''MD$Belief &may, modal-auxiliary-basic
might''MD$Belief &might, modal-auxiliary-basic
must''MD have''VB$Belief &must, modal-auxiliary-basic
probably''RB$Belief &**NOT DONE**
think''VB$Belief &think, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, X7-X9-basic,
 I5-CP-basic
thought''VBD$Belief &thought, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, X7-X9-basic,
 I5-CP-basic
thought''VBN$Belief &thought, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, X7-X9-basic,
 I5-CP-basic
thinks''VBZ$Belief &thinks, L9-basic, T1-monotransitive-basic,
 T1-passive-basicV3-passive-basic, V3-I3-basic, X7-X9-basic,
 I5-CP-basic
thinking''VBG$Belief &thinking, L9-basic, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, X7-X9-basic,
```

```
 I5-CP-basic
trust''VB$Belief &trust, T1-monotransitive-basic, T1-passive-basic,
 V3-passive-basic, V3-I3-basic, I5-CP-basic
trusted''VBD$Belief &trusted, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
trusted''VBN$Belief &trusted, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
trusts''VBZ$Belief &trusts, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
trusting''VBG$Belief &trusting, T1-monotransitive-basic,
 T1-passive-basic, V3-passive-basic, V3-I3-basic, I5-CP-basic
persuade''VB$Belief &persuade, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
persuaded''VBD$Belief &persuaded, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
persuaded''VBN$Belief &persuaded, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
persuades''VBZ$Belief &persuades, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
persuading''VBG$Belief &persuading, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic
win''VB over''IN$Belief &**NOT DONE**
won''VBD over''IN$Belief &**NOT DONE**
won''VBD over''IN$Belief &**NOT DONE**
wins''VBZ over''IN$Belief &**NOT DONE**
winning''VBG over''IN$Belief &**NOT DONE**
accept''VB$Belief &accept, T1-monotransitive-basic, T1-passive-basic,
 I5-CP-basic
accepted''VBD$Belief &accepted, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
accepted''VBN$Belief &accepted, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
accepts''VBZ$Belief &accepts, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
accepting''VBG$Belief &accepting, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
admit''VB$Belief &admit, T1-monotransitive-for-V3-verbs, V3-I3-basic,
 T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
admitted''VBD$Belief &admitted, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
admitted''VBN$Belief &admitted, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
admits''VBZ$Belief &admits, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
admitting''VBG$Belief &admitting, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
affirm''VB$Belief &affirm, T1-monotransitive-basic, T1-passive-basic,
 I5-CP-basic
affirmed''VBD$Belief &affirmed, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
affirmed''VBN$Belief &affirmed, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
affirms''VBZ$Belief &affirms, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
```

```
affirming''VBG$Belief &affirming, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
conclude''VB$Belief &conclude, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
concluded''VBD$Belief &concluded, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
concluded''VBN$Belief &concluded, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
concludes''VBZ$Belief &concludes, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
concluding''VBG$Belief &concluding, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
consider''VB$Belief &consider, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
considered''VBD$Belief &considered, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
considered''VBN$Belief &considered, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
considers''VBZ$Belief &considers, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
considering''VBG$Belief &considering, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, X7-X9-basic,
 I5-CP-basic
credit''VB$Belief &credit, T1-monotransitive-basic, T1-passive-basic
credited''VBD$Belief &credited, T1-monotransitive-basic,
 T1-passive-basic
credited''VBN$Belief &credited, T1-monotransitive-basic,
 T1-passive-basic
credits''VBZ$Belief &credits, T1-monotransitive-basic,
 T1-passive-basic
crediting''VBG$Belief &crediting, T1-monotransitive-basic,
 T1-passive-basic
postulate''VB$Belief &postulate, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
postulated''VBD$Belief &postulated, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
postulated''VBN$Belief &postulated, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
postulates''VBZ$Belief &postulates, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
postulating''VBG$Belief &postulating, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
presuppose''VB$Belief &presuppose, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
presupposed''VBD$Belief &presupposed, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
presupposed''VBN$Belief &presupposed, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
presupposes''VBZ$Belief &presupposes, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
```

```
presupposing''VBG$Belief &presupposing, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
suppose''VB$Belief &suppose, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
supposed''VBD$Belief &supposed, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
supposed''VBN$Belief &supposed, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
supposes''VBZ$Belief &supposes, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
supposing''VBG$Belief &supposing, T1-monotransitive-for-V3-verbs,
 V3-I3-basic, T1-passive-for-V3-verb, V3-passive-basic, I5-CP-basic
assume''VB$Belief &assume, T1-monotransitive-basic, T1-passive-basic,
 X7-X9-basic, I5-CP-basic
assumed''VBD$Belief &assumed, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
assumed''VBN$Belief &assumed, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
assumes''VBZ$Belief &assumes, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
assuming''VBG$Belief &assuming, T1-monotransitive-basic,
 T1-passive-basic, X7-X9-basic, I5-CP-basic
await''VB$Belief &await, T1-monotransitive-basic, T1-passive-basic
awaited''VBD$Belief &awaited, T1-monotransitive-basic,
 T1-passive-basic
awaited''VBN$Belief &awaited, T1-monotransitive-basic,
 T1-passive-basic
awaits''VBZ$Belief &awaits, T1-monotransitive-basic, T1-passive-basic
awaiting''VBG$Belief &awaiting, T1-monotransitive-basic,
 T1-passive-basic
count''VB on''IN$Belief &**NOT DONE**
counted''VBD on''IN$Belief &**NOT DONE**
counted''VBN on''IN$Belief &**NOT DONE**
counts''VBZ on''IN$Belief &**NOT DONE**
counting''VBG on''IN$Belief &**NOT DONE**
envisage''VB$Belief &envisage, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic, V3-I3-basic
envisaged''VBD$Belief &envisaged, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic, V3-I3-basic
envisaged''VBN$Belief &envisaged, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic, V3-I3-basic
envisages''VBZ$Belief &envisages, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic, V3-I3-basic
envisaging''VBG$Belief &envisaging, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic, V3-I3-basic
wait''VB for''IN$Belief &**NOT DONE**
waited''VBD for''IN$Belief &**NOT DONE**
waited''VBN for''IN$Belief &**NOT DONE**
waits''VBZ for''IN$Belief &**NOT DONE**
waiting''VBG for''IN$Belief &**NOT DONE**
presume''VB$Belief &presume, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, X7-X9-basic, I5-CP-basic
presumed''VBD$Belief &presumed, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, X7-X9-basic, I5-CP-basic
```

```
presumed''VBN$Belief &presumed, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, X7-X9-basic, I5-CP-basic
presumes''VBZ$Belief &presumes, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, X7-X9-basic, I5-CP-basic
presuming''VBG$Belief &presuming, T1-monotransitive-basic,
 T1-passive-basic, V3-I3-basic, X7-X9-basic, I5-CP-basic
suspect''VB$Belief &suspect, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
suspected''VBD$Belief &suspected, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
suspected''VBN$Belief &suspected, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
suspects''VBZ$Belief &suspects, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
suspecting''VBG$Belief &suspecting, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic

# NotBelief
doubt''VB$NotBelief &doubt, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
doubted''VBD$NotBelief &doubted, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
doubted''VBN$NotBelief &doubted, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
doubts''VBZ$NotBelief &doubts, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
doubting''VBG$NotBelief &doubting, T1-monotransitive-basic,
 T1-passive-basic, I5-CP-basic
improbable''JJ$NotBelief &**NOT DONE**
questionable''JJ$NotBelief &**NOT DONE**
uncertain''JJ$NotBelief &**NOT DONE**
unreliable''JJ$NotBelief &**NOT DONE**
unsure''JJ$NotBelief &**NOT DONE**
disbelieve''VB$NotBelief &disbelieve, T1-monotransitive-basic,
 T1-passive-basic
disbelieved''VBD$NotBelief &disbelieved, T1-monotransitive-basic,
 T1-passive-basic
disbelieved''VBN$NotBelief &disbelieved, T1-monotransitive-basic,
 T1-passive-basic
disbelieves''VBZ$NotBelief &disbelieves, T1-monotransitive-basic,
 T1-passive-basic
disbelieving''VBG$NotBelief &disbelieving, T1-monotransitive-basic,
 T1-passive-basic
distrust''VB$NotBelief &distrust, T1-monotransitive-basic,
 T1-passive-basic
distrusted''VBD$NotBelief &distrusted, T1-monotransitive-basic,
 T1-passive-basic
distrusted''VBN$NotBelief &distrusted, T1-monotransitive-basic,
 T1-passive-basic
distrusts''VBZ$NotBelief &distrusts, T1-monotransitive-basic,
 T1-passive-basic
distrusting''VBG$NotBelief &distrusting, T1-monotransitive-basic,
 T1-passive-basic
discount''VB$NotBelief &discount, T1-monotransitive-basic,
```

```
 T1-passive-basic
discounted''VBD$NotBelief &discounted, T1-monotransitive-basic,
 T1-passive-basic
discounted''VBN$NotBelief &discounted, T1-monotransitive-basic,
 T1-passive-basic
discounts''VBZ$NotBelief &discounts, T1-monotransitive-basic,
 T1-passive-basic
discounting''VBG$NotBelief &discounting, T1-monotransitive-basic,
 T1-passive-basic
discredit''VB$NotBelief &discredit, T1-monotransitive-basic,
 T1-passive-basic
discredited''VBD$NotBelief &discredited, T1-monotransitive-basic,
 T1-passive-basic
discredited''VBN$NotBelief &discredited, T1-monotransitive-basic,
 T1-passive-basic
discredits''VBZ$NotBelief &discredits, T1-monotransitive-basic,
 T1-passive-basic
discrediting''VBG$NotBelief &discrediting, T1-monotransitive-basic,
 T1-passive-basic
unbelieving''JJ$NotBelief &**NOT DONE**

# Negation
not''RB$Negation &not,
no''RB$Negation &**NOT DONE**
none''NN$Negation &**NOT DONE**
never''RB$Negation &never,
n't''RB$Negation &n't,
```

Appendix B

Mapping LDOCE Codes to Subcategorization Codes

This appendix provides a listing of the rules used to map between LDOCE codes and subcategorization codes. For example, the LDOCE code T3 or D3 or V3 indicates that the two templates V3-passive-basic and V3-I3-basic are applicable. Exceptions two the initial LDOCE codes are given in the second part of this appendix.

```
-- Transitive cases --

If (T3 or D3 or V3)
Then V3-passive-basic, V3-I3-basic

If (T1 and D1)
Then T1-monotransitive-for-D1-verbs

If (T1 and V3)
Then T1-monotransitive-for-V3-verbs, T1-passive-for-V3-verb

If D1
Then D1-ditransitive-basic

If T1 and (not D1 or V3)
Then T1-monotransitive-basic, T1-passive-basic

If (T5 or T6) // CP[+wh] & [He contemplated [whether to go ...]
Then I5-CP-basic

If T4 // VP[+prog] & [He denied [eating food]]
Then V3-I3-basic

If (X7 or X9) // NP AP & [He wished [him] [dead]]
Then X7-X9-basic

If V2 // NP VP[+inf] & [He made him [eat ...]]
Then V3-I3-basic

If L9 // ADJ & [He appeared [stupid]]
Then L9-basic
```

```
-- Other cases --

If F3
Then JJ-infinitive

If WV2
Then modal-auxiliary-basic

-- Now for the intransitive cases --

If I-TO // PP[to]  & [He aspired to ...]
Then V3-I3-basic

If (I3 and T3) // verb like "fail"
Then I-intransitive-for-I-in-and-I3-verb, V3-I3-basic,
     I-in-intransitive-basic, I-in-intransitive-VBG

Else if (I and L9) // verb like "flop"
Then I-intransitive-basic

Else (I-in or I-for or I-with or I-after or I-among or I-over
           or I-on or I-from or I-against or I-of)
Then I-Prep-intransitive-basic, I-Prep-intransitive-VBG

Else If I3
Then V3-I3-basic

Else If I // not one of the more elaborate cases above
Then I-intransitive-basic
```

Most intransitive LDOCE codes were not applicable to modality constructions. For example, *hunger* (in the *Want* modality class) has a modal reading of "desire" when combined with the preposition *for* (as in *she hungered for a promotion*), but not in its pure intransitive form (e.g., *he hungered all night*). Thus the LDOCE code `I` associated with the verb *hunger* was hand-changed to `I-FOR` for our purposes. There were 43 such exceptions, listed below. Once the LDOCE codes were hand-verified (and modified accordingly), the mapping above was applied to produce the corresponding subcategorization codes.

```
-- Exceptions to LDOCE codes --

abort: delete I
abstain: delete I and add I-FROM
accept: delete I
admit: delete I
affirm: delete I
aim: delete I
appear: delete I
arrive: delete I and add I-AT
aspire: delete I and add V3
attack: delete I
believe: delete I
conclude: delete I
consider: delete I
contemplate: delete I
```

```
contend: delete I
covet: delete I
design: delete I
despair: delete I and add I-OF
entreat: delete I
fancy: delete I
fear: delete I
hunger: delete I and add I-FOR
indicate: delete I
know: delete I
need: delete I
order: delete I
pass: delete I and add T1-ON
permit: delete I
pine: delete I and add I-FOR
prevail: delete I and add I-AGAINST, I-IN
reach: delete I
satisfy: delete I
signify: delete I
solicit: delete I
strive: delete I
strive: delete I and add L9, I-WITH
succeed: delete I
tackle: delete I
think: delete I
trust: delete I
win: delete I
work: delete I and add I-AT, I-ON
wrangle: delete I and add I-WITH
```

Appendix C

Urdu Tokenization

We developed a tokenizer and word normalizer for Urdu to ensure that all of our processes were operating on the same token stream. The tokenization/normalization unit had three requirements:

1. Any standoff annotation files that exist for the file being normalized must be updated to ensure that the character offsets and text snippets they contain are updated to reflect any changes to the underlying text.

2. To the greatest extent possible, tokenization must precede normalization. Put another way, the normalization phase must include as many of the desired rewrite rules as possible, but it may not change the number of tokens by adding or removing white space. This constraint is aimed at allowing annotators to operate on tokenized text that is as close to the original text as possible.

3. Normalization must respect SGML markup contained in the document, but must not fail if that markup is ill-formed. In general, when a document contains SGML markup we should not try to normalize the content of the SGML tags, only the text portions of the document. However, files that do not contain markup may contain characters that look like markup (e.g., "¡" characters). The tokenizer and normalizer must be able to handle either case.

Together, these constraints dictated a two-stage process. Phase one is tokenization. The result of the tokenization phase is a version of the text in which each whitespace-delimited character sequence is a token. Phase Two is normalization. The normalization phase produces the final version of the text in which all normalization transformations (e.g., substitution of one character for another) have been made. Each phase is specified by a sequence of rule sets. Each rule in a rule set comprises a regular expression as the lefthand side, and a replacement string as the righthand side.

To meet Criterion 1, each standoff annotation file is tokenized and normalized at the same time as its underlying text file. To ensure that each pointer into the original text file is properly mapped onto the tokenized/normalized file, transformation rules are not applied across any pointer into the text found in any annotation file. The software tracks the offset of each character in the file as transformations are applied. Once each rule set has been applied and the resultant text has been written to output, the standoff annotation files are modified. Two kinds of modifications are made. First, any offsets into the underlying text file are updated to indicate the new position of the text referred to. Second, if the annotation file contains extracted snippets of text, those snippets are updated with the corresponding transformed text.

To meet criterion 2, the normalization phase never changes the number of whitespace-delimited tokens in the text. As a safety check, the normalizer will die with an appropriate error message if the number of tokens does for some reason change. To ensure that all appropriate tokenization rules are applied, each such tokenization rule must have the appropriate normalization rules applied to it. For example, consider the following tokenization rule:

```
"\\b\x{06C1}\x{0648}\x{06AF}\x{0627}\\b" =>
    "\x{06C1}\x{0648} \x{06AF}\x{0627}"
```

This rule inserts a space between the second and third characters of a four-character sequence (thus introducing a new token). Suppose there is also a normalization rule that maps a character onto one of the characters of the lefthand side of this rule, such as:

```
"\x{FEEE}" =>
   "\x{0648}"
```

To ensure that the tokenization rule is applied before the normalization rule, we must introduce a new tokenization rule:

```
"\\b\x{06C1}\x{FEEE}\x{06AF}\x{0627}\\b" =>
   "\x{06C1}\x{FEEE} \x{06AF}\x{0627}"
```

This rule enforces the new tokenization without applying the character normalization, which will be applied during the subsequent normalization phase. We have automated the process of applying a set of normalization rules to a set of tokenization rules in this way; this allows tokenization rules to be written without consideration of all normalization rules that might be applied.

To meet criterion 3, we allow the user to specify whether the files being normalized contain SGML markup, and if so, which tag delimits text to be operated upon. A simple finite state machine determines which portions of the text are within the specified tag, and restricts processing to that text. If the user indicates that no markup is present, all text is processed.

References

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 16–23, Athens, Greece, March. Association for Computational Linguistics.

Anthony Ades and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

Johan Van Der Auwera and Andreas Ammann. 2005. Overlap between situational and epistemic modal marking. In Martin Haspelmath, Matthew S. Dryer, David Gil, and Bernard Comrie, editors, *World Atlas of Language Structures*, chapter 76, pages 310–313. Oxford University Press.

Jason Baldridge and Miles Osborne. 2008. Active learning and logarithmic opinion pools for hpsg parse selection. *Nat. Lang. Eng.*, 14(2):191–222.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, Michigan.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Michael Bloodgood and K Vijay-Shanker. 2009a. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 39–47, Boulder, Colorado, June. Association for Computational Linguistics.

Michael Bloodgood and K Vijay-Shanker. 2009b. Taking into account the differences between actively and passively acquired data: The case of active learning with support vector machines for imbalanced datasets. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 137–140, Boulder, Colorado, June. Association for Computational Linguistics.

I. Borg and P. Groenen. 1997. *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, New York.

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

Jaime Carbonell, S Klein, D Miller, M Steinbaum, T Garssiany, and J Frey. 2006. Context-based machine translation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA-2006)*, Cambridge, Massachusetts.

Colin Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 72–80, Columbus, Ohio, June. Association for Computational Linguistics.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Brooke Cowan, Ivona Kučerová, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 232–241, Sydney, Australia, July. Association for Computational Linguistics.

Steve DeNeefe and Kevin Knight. 2009. Synchronous tree adjoining machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 727–736, Singapore, August. Association for Computational Linguistics.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural*

Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 755–763, Prague, Czech Republic, June. Association for Computational Linguistics.

John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, Prague, Czech Republic.

Bonnie Dorr, Mona Diab, Lori Levin, Teruko Mitamura, Christine Piatko, and Owen Rambow. 2009. Modality/polarity annotation guidelines.

Christopher Dyer, Aaron Cordova, Alex Mont, and Jimmy Lin. 2009. Fast, easy, and cheap: Construction of statistical machine translation models with mapreduce. In *Proceedings of the Workshop on Statistical Machine Translation*, Athens, Greece.

Pascale Fung and Lo Yuen Yee. 1998. An ir approach for translating new words from nonparallel, comparable texts. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-CoLing-1998)*, Montreal, Canada.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American chapter of the Association for Computational Linguistics (HLT/NAACL-2004)*, Boston, Massachusetts.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL-CoLing-2006)*, Sydney, Australia.

Nikesh Garera, Chris Callison-Burch, and David Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, Boulder, Colorado.

Ben Hachey, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 144–151, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Robbie Haertel, Eric Ringger, Kevin Seppi, James Carroll, and Peter McClanahan. 2008. Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 65–68, Columbus, Ohio, June. Association for Computational Linguistics.

Gholamreza Haffari and Anoop Sarkar. 2009. Active learning for multilingual statistical machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 181–189, Suntec, Singapore, August. Association for Computational Linguistics.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. 2009. Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 415–423, Boulder, Colorado, June. Association for Computational Linguistics.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Colmbus, Ohio.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the International Workshop on Parsing Technologies*, Vancouver, BC, Canada.

Thorsten Joachims. 2005. A support vector method for multivariate performance measures. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 377–384, New York, NY, USA. ACM.

Jae Dong Kim, Ralf D. Brown, Peter J. Jansen, and Jaime G. Carbonell. 2005. Symmetric probabilistic

alignment for example-based translation. In *Proceedings of the Tenth Workshop of the European Association for Machine Translation (EAMT-05)*, May.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*.

Philipp Koehn and Kevin Knight. 2000. Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm. In *Proceedings of AAAI*, Philadelphia, Pennsylvania.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *ACL 2002 Workshop on Unsupervised Lexical Acquisition*, Philadelphia, Pennsylvania.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American chapter of the Association for Computational Linguistics (HLT/NAACL-2003)*, Edmonton, Alberta.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the 6th Biennial Conference of the Association for Machine Translation in the Americas (AMTA-2004)*, Washington DC.

Philipp Koehn. 2005. A parallel corpus for statistical machine translation. In *Proceedings of MT-Summit*, Phuket, Thailand.

Angelika Kratzer. 2009. Plenary Address at the Annual Meeting of the Linguistic Society of America.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American chapter of the Association for Computational Linguistics (HLT/NAACL-2004)*, Boston, Massachusetts.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March. Association for Computational Linguistics.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment templates for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL-CoLing-2006)*, Sydney, Australia.

Adam Lopez. 2007. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic.

Zhiliang Ma, Adam Cardinal-Stakenas, Youngser Park, and Carey E. Priebe. 2008. Combining dissimilarity representations in embedding product space. In *Proceedings of the Interface 2008*, May.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. Statistical machine translation with syntactified target language phrases. In *In EMNLP*, pages 44–52.

Dan Melamed. 2004. Statistical machine translation by parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004)*, Barcelona, Spain.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio, June. Association for Computational Linguistics.

Scott Miller, Heidi J. Fox, Lance A. Ramshaw, and Ralph M. Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of Applied Natural Language Processing and the North American Association for Computational Linguistics*.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Nirenburg and McShane. 2008. The formulation of modalities (speaker attitude) in ontosem.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, Philadelphia, Pennsylvania.

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

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, Philadelphia, Pennsylvania.

Aaron B. Phillips. 2007. Sub-phrasal matching and structural templates in Example-Based MT. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, September.

Aaron Phillips. 2009. Cunei machine translation. [Online:accessed 19-August-2009].

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *33rd Annual Meeting of the Association for Computational Linguistics (ACL-1995)*, Cambridge, Massachusetts.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, College Park, Maryland.

Charles Schafer and David Yarowsky. 2002a. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Conference on Natural Language Learning-2002*, pages 146–152.

Charles Schafer and David Yarowsky. 2002b. Inducing translation lexicons via diverse similarity measures and bridge languages. In *COLING-02: proceedings of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Greg Schohn and David Cohn. 2000. Less is more: Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning*, pages 839–846. Morgan Kaufmann, San Francisco, CA.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL 2006)*, pages 787–794.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA-2006)*, Cambridge, Massachusetts.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER? exploring different human judgments with a tunable mt metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation at the 12th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2009)*, Athens, Greece.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, September.

Christoph Tillmann. 2003. A projection extension algorithm for statistical machine translation. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, Sapporo, Japan.

M. W. Trosset and C. E. Priebe. 2008. The out-of-sample problem for classical multidimensional scaling. *Computational Statistics and Data Analysis*, 52(10):4635–4642.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79, Sapporo, Japan, July. Association for Computational Linguistics.

Ashish Venugopal and Andreas Zollmann. 2009. Grammar based statistical MT on Hadoop: An end-to-end toolkit for large scale PSCFG based MT. *Prague Bulletin of Mathematical Linguistics*, 91.

Ashish Venugopal, Stephan Vogel, and Alex Waibel. 2003. Effective phrase translation extraction from alignment models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, Sapporo, Japan.

Andreas Vlachos. 2008. A stopping criterion for active learning. *Computer Speech and Language*, 22(3):295–312.

Stephan Vogel. 2005. Pesa: phrase pair extraction as sentence splitting. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 251–258.

Kai von Fintel and Sabine Iatridou. 2009. Morphology, syntax, and semantics of modals. Lecture notes for 2009 LSA Institute class.

Dekai Wu and Pascale Fung. 2005. Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP-2005)*, Jeju Island, Korea.

Christopher C. Yang and Kar Wing Li. 2003. Automatic construction of english/chinese parallel corpora. *Journal of the American Society for Information Science and Technology*, 54(8):730–742.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, Ohio, June. Association for Computational Linguistics.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June. Association for Computational Linguistics.

DRAFT: NOT FOR PUBLIC RELEASE
PUBLIC RELEASE PENDING
