Inductive Detection of Language Features via Clustering Minimal Pairs: Toward Feature-Rich Grammars in Machine Translation

Jonathan H. Clark, Robert Frederking, Lori Levin
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{jhclark,ref,lsl}@cs.cmu.edu

Abstract

Syntax-based Machine Translation systems have recently become a focus of research with much hope that they will outperform traditional Phrase-Based Statistical Machine Translation (PBSMT). Toward this goal, we present a method for analyzing the morphosyntactic content of language from an Elicitation Corpus such as the one being including in the LDC’s LCTL language packs. The presented method discovers a mapping between morphemes and linguistically relevant features. By providing this tool with which structure-based models of MT can be augmented with these rich features, we believe the discriminative power of current models can be improved. We conclude by outlining how the resulting output can then be used in inducing a morphosyntactically feature-rich grammar for AVENUE, a modern syntax-based MT system.

1 Introduction

Recent trends in Machine Translation have begun moving toward the incorporation of syntax and structure in translation models in hopes of gaining better translation quality. In fact, some structure-based systems have already shown that they can outperform phrase-based SMT systems (Chiang, 2005). Still, even the best data-driven systems have not fully explored the depth of such linguistic features as morphosyntax.

Certainly, many have brought linguistically motivated features into their models in the past. Huang and Knight (2006) explored relabeling of non-terminal symbols to embed more information directly into the backbone of the grammar. Bonneau-Maynard et al. (2007) argue that incorporation of morphosyntax in the form of a part of speech (POS) language model can improve translation. While these approaches do make use of various linguistic features, we have only begun to scratch the surface of what actually occurs in the languages of the world. We wish to address such issues as case marking, subject-verb agreement, and numeral-classifier agreement by providing models with information about which morphemes correspond to which grammatical meanings.

2 Task Overview

Feature Detection is the process of determining from a corpus annotated with feature structures (Figure 2) which feature values (Figure 1) have a distinct representation in a target language in terms of morphemes (Figure 3). By leveraging knowledge from the field of language typology, we know what types of phenomena are possible across languages and, thus, which features to include in our feature specification.

But not every language will display each of these phenomena. Our goal is to determine which feature values (e.g. singular, dual, plural) have a distinct encoding in a given target language. Viewed differently, we can ask which feature values can be clustered by similarity. For instance, in Chinese, we would expect singular, plural and dual to be members of the same cluster while for Arabic we should place each of these into separate clusters to indicate
As seen in the Table 1, the input sentences are extracted from the source language and translated to the target language. Each sentence is accompanied by the feature structure that describes the grammatical properties of the sentence.

They are each grammaticalized differently. Similarly, English would have two clusters for the feature number: (singular) and (dual, plural). Further, we would like to determine which morphemes express each of these values (or value clusters). For example, English expresses negation with the morphemes *no* and *not*, whereas questions are expressed by reordering of the auxiliary verb or the addition of a wh-word.

Though many modern corpora contain feature-annotated utterances, these corpora are often not suitable for feature detection. For this purpose, we use an Elicitation Corpus (see Figure 2), a corpus that has been carefully constructed to provide a large number of minimal pairs of sentences such as *He sings* and *She sings* so that only a single feature (e.g. gender) differs between the two sentences. Also, notice that the feature structures are sometimes more detailed than the source language sentence. For example, English does not express dual number, but we might want to include this feature in our Elicitation Corpus (especially for a language such as Arabic). For these cases, we include a context field for the translator with an instruction such as “Translate this sentence as if there are two girls.”

In the past, we proposed deductive (rule-based) methods for feature detection. In this paper, we propose the use of inductive feature detection, which operates directly on the feature set that the corpus has been annotated with, removing the need for manually written rules. We define inductive feature detection as a recall-oriented task since its output is intended to be analyzed by a Morphosyntactic Lexicon Generator, which will address the issue of precision. This, in turn, allows us to inform a rule learner about which language features can be clustered and handled by a single set of rules and which must be given special attention. However, due to the complexity of this component, describing it is beyond the scope of this paper. We also note that future work will include the integration of a morphology analysis system such as ParaMor (Monson et al., 2007) so that we can extract and annotate the valuable morphosyntactic information of inflected languages. An example of this processing pipeline is given in Fig-

### Figure 1: An example feature specification.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Feature Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>np-gen</td>
<td>m, f, n</td>
<td>Biological Gender</td>
</tr>
<tr>
<td>np-def</td>
<td>+, -</td>
<td>Definiteness</td>
</tr>
<tr>
<td>np-num</td>
<td>sg, dl, pl</td>
<td>Number</td>
</tr>
<tr>
<td>c-ten</td>
<td>past, pres, fut</td>
<td>Tense</td>
</tr>
<tr>
<td>np-function</td>
<td>act, und</td>
<td>Actor and undergoer participant roles</td>
</tr>
<tr>
<td>c-function</td>
<td>main, rel</td>
<td>Main and relative clause roles</td>
</tr>
</tbody>
</table>

### Figure 2: An example of sentences that might be found in an elicitation corpus. Notice that each sentence differs from some other sentence in the corpus by exactly one feature value. This enables us to see how the written form of the language changes (or does not change) when the grammatical meaning changes.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
<th>Candidate Morphemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>np-gen</td>
<td>m</td>
<td>el, niño</td>
</tr>
<tr>
<td>np-gen</td>
<td>f</td>
<td>ella, niña</td>
</tr>
<tr>
<td>np-gen</td>
<td>n</td>
<td><em>unobserved</em></td>
</tr>
<tr>
<td>np-def</td>
<td>+</td>
<td>el, la, las</td>
</tr>
<tr>
<td>np-def</td>
<td>-</td>
<td>una, unas</td>
</tr>
<tr>
<td>np-num</td>
<td>sg</td>
<td>el, ella, la, una, come, niño, niña</td>
</tr>
<tr>
<td>np-num</td>
<td>dl-pl</td>
<td>las, unas, comen, niñas</td>
</tr>
<tr>
<td>c-ten</td>
<td>past-pres</td>
<td>–</td>
</tr>
<tr>
<td>c-ten</td>
<td>fut</td>
<td><em>unobserved</em></td>
</tr>
</tbody>
</table>

Figure 3: An example of the output of our system for the above corpus: a list of feature-morpheme pairings.

Figure 4: An outline of the steps from an input Elicitation Corpus to the application of a morphosyntactically feature rich grammar in a MT decoder. This paper discusses the highlighted inductive feature detection component. Note that this is just one possible configuration for integrating inductive feature detection into system training.

3 The Need to Observe Real Data

One might argue that such information could be obtained from a grammatical sketch of a language. However, these sketches often focus on the “interesting” features of a language, rather than those that are most important for machine translation. Further, not all grammatical functions are encoded in the elements that most grammatical sketches focus on. According to Construction Grammar, such information is also commonly found in constructions (Kay, 2002). For example, future tense is not grammaticalized in Japanese according to most reference sources, yet it may be expressed with a construction such as watashi wa gakoo ni iku yode desu (lit. “I have a plan to go to school.”) for I will go to school. Feature detection informs us of such constructionalized encodings of language features for use in improving machine translation models.

Recognizing the need for this type of data, the LDC has included an Elicitation Corpus in their Less Commonly Taught Languages (LCTL) language packs (Simpson et al., 2008). Already, these language packs have been translated into Thai, Bengali, Urdu, Hungarian, Punjabi, Tamil, and Yoruba. With structured elicitation corpora already being produced on a wide scale, there exists plenty of data that can be exploited via feature detection. These language packs will start being released to the general research community this year through LDC’s catalog.

4 Applications

4.1 Induction of Feature-Rich Grammars

Given these outputs, a synchronous grammar induction system can then use these feature-annotated morphemes and the knowledge of which features are expressed to create a feature rich grammar. Consider the example in Figure 5, which shows Urdu subject-verb agreement taking place while being separated by 12 words. Traditional n-gram Language Models (LM’s) would not be able to detect any disagreements more than n words away, which is the normal case for a trigram LM. Even most syntax-based systems would not be able to detect this problem without using a huge number of non-terminals, each marked for all possible agreements. A syntax-based
system might be able to check this sort of agreement if it produced a target-side dependency tree as in Ding and Palmer (2005). However, we are not aware of any systems that attempt this. Therefore, the correct hypotheses, which have correct agreement, will likely be produces as hypotheses of traditional beam-search MT systems, but their features might not be able to discern the correct hypothesis, allowing it to fall below the 1-best or out of the beam entirely. By constructing a feature-rich grammar in a framework that allows unification-based feature constraints such as AVENUE (Carbonell et al., 2002), we can prune these bad hypotheses having disagreement from the search space.

Returning to the example of subject-verb agreement, consider the following Urdu sentences taken from the Urdu-English Elicitation Corpus in LDC’s LCTL language pack:

Danish ne Amna ko sza di
Danish ERG Amna DAT punish give.PERF
“Danish punished Amna.”
Danish Amna ko sza dita hai
Danish Amna DAT punish give.HAB be.PRES
“Danish punishes Amna.”

These examples show the split-ergativity of Urdu in which the ergative marker ne is used only for the subject of transitive, perfect aspect verbs. In particular, since these sentences have the perfect aspect marked on the light verb di, a closed class (P ornima and Koenig, 2008), feature detection will allow the induction of a grammar that percolates a feature up from the VP containing di indicating that its aspect is perfect. Likewise, the NP containing Danish ne will percolate a feature up indicating that the use of ne requires perfect aspect. If, during translation, a hypothesis is proposed that does not meet either of these conditions, unification will fail and the hypothesis will be pruned.

Certainly, unification-based grammars are not the only way in which this rich source of linguistic information could be used to augment a structure-based translation system. One could also imagine a system in which the feature annotations are simply used to improve the discriminative power of a model. For example, factored translation models (Koehn and Hoang, 2007) retain the simplicity of phrase-based SMT while adding the ability to incorporate additional features. Similarly, there exists a continuum of degrees to which this linguistic information can be used in current syntax-based MT systems. As modern systems move toward integrating many features (Liang et al., 2006), resources such as this will become increasingly important in improving translation quality.

5 System Description

In the following sections, we will describe the process of inductive feature detection by way of a running example.

5.1 Feature Specification

The first input to our system is a feature specification (Figure 1). The feature specification used for this experiment was written by an expert in language typology and is stored in a human-readable XML format. It is intended to cover a large number of phenomena that are possible in the languages of the world. Note that features beginning with np− are participant (noun) features while features beginning with c− are clause features. The feature specification allows us to know which values are unobserved from the Elicitation Corpus. The definitions of the first four features and their values are used so we still know about values that might not have been observed. The last two function features and their values tell us what possible roles participants and clauses can take in sentences.
5.2 Elicitation Corpus

As outlined in Section 3, feature detection uses an Elicitation Corpus (see Figure 2), a corpus that has been carefully constructed to provide a large number of minimal pairs of sentences such as *He sings* and *She sings* so that only a single feature (e.g., gender) differs between the two sentences (Levin et al., 2006; Alvarez et al., 2006). If two features had varied at once (e.g., *It sang*) or lexical choice varied (e.g., *She reads*), then making assertions about which features the language does and does not express becomes much more difficult.

Notice that each input sentence has been tagged with an identifier for a lexical cluster as a pre-processing step. Specifying lexical clusters ensures that we don’t compare sentences with different content just because their feature structures match. For example, we would not want to compare *The car raced the train* and *The train raced the car* nor *The student snored* and *The professor snored*.

Though any feature-annotated corpus can be used in feature detection, the amount of useful information extracted from the corpus is directly dependent on how many minimal pairs can be formed from the corpus. For instance, one might consider using a morphologically annotated corpus or even an automatically parsed corpus in place of the elicitation corpus. Even though these resources are likely to suffer from having very sparse minimal pairs due to their uncontrolled usage of vocabulary, they might still contain some amount of useful information. However, since we seek both to apply these methods to language for which there are currently no manually annotated corpora and to investigate features that existing parsers generally cannot identify (e.g., generic nouns and evidentiality), we will not mention these types of resources any further.

5.3 Minimal Pair Clustering

Minimal pair clustering is the process of grouping all possible sets of minimal pairs, those pairs of sentences that have exactly one difference between their feature structures. We use wildcard feature structures to represent each minimal pair cluster. We define a wildcard feature as any feature whose value is *, which denotes that the value matches another * rather than its original feature value. Similarly, we define the feature context of the wildcard feature be the enclosing participant and clause type for a np-feature or the enclosing clause for a c-type feature. Then, for each sentence s in the corpus, we substitute a wildcard feature for each of the values v in its feature structure, and we append s to the list of sentences associated with this wildcard feature structure. A sample of some of the minimal pairs for our running example are shown in Figure 6.

Here, we show minimal pairs for just one wildcard, though multiple wildcards may be created if one wishes to examine how features interact with one another. This could be useful in cases such as Hindi where the perfective verb aspect interacts with the past verb tense and the actor NP function to add the case marker *ne* (for split ergativity of Urdu, see Section 4.1). That said, a downstream component such as a Morphosyntactic Lexicon Generator would perhaps be better suited for the analysis of feature interactions. Also, note that the feature context is not used when there is only one wildcard feature. The feature context becomes useful when multiple wildcards are added in that it may also act as a wildcard feature.

The next step is to organize the example sentences into a table that helps us decide which examples can be compared and stores information that will inform our comparison. Briefly, any two sentences belonging to the same minimal pair cluster and lexical cluster will eventually get compared. As specified in Algorithm 1, we create a table like that in Figure 7. Having collected this information, we are now ready to begin clustering feature values.

**Algorithm 1 Organize()**

Require: Minimal pairs, lexical clusters, and the feature specification.

Ensure: A table T of comparable examples.

for all pair m ∈ minimalPairs do
  for all sentence s ∈ m do
    f ← wildcardFeature(s, m)
    v ← featureValue(s, f)
    c ← featureContext(m)
    ℓ ← lexCluster(s)
    T[f, m, c, ℓ, v] ← T[f, m, c, ℓ, v]∪ s
  return T
<table>
<thead>
<tr>
<th>ID</th>
<th>Set Members</th>
<th>Feature</th>
<th>Feature Context</th>
<th>Feature Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>{s1, s2}</td>
<td>np-gen</td>
<td>((act))</td>
<td>((act (np-gen *)) (np-num sg) (np-def +))</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(und (np-gen f) (np-num sg) (np-def +)) (c-ten pres))</td>
</tr>
<tr>
<td>m2</td>
<td>{s1, s3}</td>
<td>np-ten</td>
<td>()</td>
<td>((act (np-gen m) (np-num sg) (np-def +))</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(und (np-gen f) (np-num sg) (np-def +)) (c-ten *))</td>
</tr>
<tr>
<td>m3</td>
<td>{s4, s5, s7, s8}</td>
<td>np-gen</td>
<td>((act))</td>
<td>((act (np-gen *) (np-num sg) (np-def +)) (c-ten pres))</td>
</tr>
<tr>
<td>m4</td>
<td>{s5, s7, s8}</td>
<td>np-num</td>
<td>((act))</td>
<td>((act (np-gen f) (np-num *) (np-def *)) (c-ten pres))</td>
</tr>
<tr>
<td>m5</td>
<td>{s6, s9}</td>
<td>np-num</td>
<td>((act))</td>
<td>((act (np-gen f) (np-num *) (np-def -)) (c-ten pres))</td>
</tr>
<tr>
<td>m6</td>
<td>{s5, s6}</td>
<td>np-def</td>
<td>((act))</td>
<td>((act (np-gen f) (np-num sg) (np-def *)) (c-ten pres))</td>
</tr>
<tr>
<td>m7</td>
<td>{s7, s9}</td>
<td>np-def</td>
<td>((act))</td>
<td>((act (np-gen f) (np-num pl) (np-def *)) (c-ten pres))</td>
</tr>
</tbody>
</table>

Figure 6: An example subset of minimal pairs that can be formed from the corpus in Figure 2.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Min. Pair</th>
<th>Feature Context</th>
<th>Lex. Cluster</th>
<th>Feature Value.</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>np-gen</td>
<td>m1</td>
<td>((act))</td>
<td>(\ell 1)</td>
<td>m</td>
<td>s1</td>
</tr>
<tr>
<td>np-gen</td>
<td>m1</td>
<td>((act))</td>
<td>(\ell 1)</td>
<td>f</td>
<td>s2</td>
</tr>
<tr>
<td>np-ten</td>
<td>m2</td>
<td>()</td>
<td>(\ell 1)</td>
<td>pres</td>
<td>s1</td>
</tr>
<tr>
<td>np-ten</td>
<td>m2</td>
<td>()</td>
<td>(\ell 1)</td>
<td>past</td>
<td>s3</td>
</tr>
<tr>
<td>np-num</td>
<td>m4</td>
<td>((act))</td>
<td>(\ell 2)</td>
<td>sg</td>
<td>s5</td>
</tr>
<tr>
<td>np-num</td>
<td>m4</td>
<td>((act))</td>
<td>(\ell 2)</td>
<td>pl</td>
<td>s7</td>
</tr>
<tr>
<td>np-num</td>
<td>m4</td>
<td>((act))</td>
<td>(\ell 2)</td>
<td>dl</td>
<td>s8</td>
</tr>
<tr>
<td>np-num</td>
<td>m5</td>
<td>((act))</td>
<td>(\ell 2)</td>
<td>sg</td>
<td>s6</td>
</tr>
<tr>
<td>np-num</td>
<td>m5</td>
<td>((act))</td>
<td>(\ell 2)</td>
<td>pl</td>
<td>s9</td>
</tr>
</tbody>
</table>

etc.

Figure 7: An example subset of the organized items that can be formed from the minimal pairs in Figure 6. Each of these items that has a matching minimal pair ID, feature context, and lexical cluster ID can be compared during feature detection.
5.4 Feature Value Clustering

During the process of feature value clustering, we collapse feature values that do not have a distinct encoding in the target language into a single group. This is helpful both as information to components using the output of inductive feature detection and later as a method of reducing data sparseness when creating morpheme-feature pairings. We represent the relationship between the examples we have gathered for each feature as a feature expression graph (FEG). We define a feature expression graph (FEG) for a feature as a graph on \(|v|\) vertices where \(v\) is the number of possible values of \(f\) (though for most non-trivial cases, it is more conveniently represented as a triangular matrix).

Each vertex of the FEG corresponds to a feature value (e.g., singular, dual) while each arc contains the list of examples that are comparable according to the table from the previous step. The examples at each arc are organized into those that had the same target language string, indicating that the feature values are not distinctly expressed, and those that had a different target language string, indicating that the change in grammatical meaning represented in the feature structure has a distinct encoding in the target language. Algorithm 2 more formally specifies the creation of a FEG. The FEG’s for our running example are shown in Figure 8. From these statistics, we then estimate the maximum likelihood probability of each feature value pair being distinctly encoded as shown in Figure 9.

The interpretation of these probabilities may not be completely obvious. They estimate the likelihood of a language encoding a given feature given that the meaning of that feature is intended to be conveyed. These probabilities should not be interpreted as a traditional likelihood of encountering a given lexical item.

Finally, we cluster by randomly selecting a starting vertex for a new cluster and adding vertices to that cluster, following arcs out from the cluster that have a weight lower than some threshold \(\theta\). When no more arcs may be followed, a new start vertex is selected and another cluster is formed. This is repeated until all feature values have been assigned to a cluster. For our running example, we use \(\theta = 0.6\), which results in the following clusters being formed:

Algorithm 2 Collecting statistics for each FEG.

Require: The table \(T\) from the previous step.
Ensure: A complete graph as an arc list with the observed similarities and differences for each feature value.

\[
\text{for all } \quad \forall i, j \in T \text{ s.t. } (m_i, c_i, \ell_i) = (m_j, c_j, \ell_j) \quad \text{do}
\]

\[
(v_i, v_j) \leftarrow (\text{featureValue}(s_i), \text{featureValue}(s_j))
\]

\[
\text{if tgt}(s_i) = tgt(s_j) \text{ then}
\]

\[
\arcs[v_i, v_j] \leftarrow \arcs[v_i, v_j] \cup (s_i, s_j, m,EQ)
\]

\[
\text{else}
\]

\[
\arcs[v_i, v_j] \leftarrow \arcs[v_i, v_j] \cup (s_i, s_j, m,NEQ)
\]

return \(\arcs\)

5.5 Morpheme-Feature Pairing

Finally, using the information from above about which values should be examined as a group and which sentence pairs exemplify an orthographic difference, we examine each pair of target language sentences to determine which words changed to reflect the change in grammatical meaning. This process is outlined in Algorithm 3. The general idea is that for each arc going out of a feature value vertex we examine all of the target language sentence pairs that expressed a difference. We then take the words that were in the vocabulary of the target sentence for the current feature value, but not in the sentence it was being compared to and add them to the list of words that could be used to express this feature value.

6 Evaluation and Results

We evaluated the output of feature detection with one wildcard feature as applied to the Elicitation Corpus from the LDC’s Urdu-English LCTL language pack. Threshold parameters were set to small values \((\theta = 0.05)\). Note that an increase in precision might be possible by tuning this value; however, as stated, we are most concerned with recall.

An initial attempt was made to create a gold standard against which recall could be directly calculated. However, the construction of this gold stan-
Algorithm 3 Determine which morphemes are associated with which feature values.

Require: List of clusters $C$ and list of FEGs $F$

Ensure: A list of morphemes associated with each feature value

for all feature $\in F$ do
  for all vertex $\in$ feature do
    for all arc $\in$ vertex do
      for all $(s_1, s_2, m, NEQ) \in$ arc do
        $v_1 \leftarrow$ featureValue($s_1, m$)
        $v_2 \leftarrow$ featureValue($s_2, m$)
        if $v_1 \neq v$ then $(s_1, v_1) \leftrightarrow (s_2, v_2)$
        $w_1 \leftarrow$ vocabulary($s_1$)
        $w_2 \leftarrow$ vocabulary($s_2$)
        $\delta \leftarrow W_1 - W_2$
      for all $w \in$ freq do
        freq[$w$]++
      for all $w \in$ freq do
        $p = freq[w] / \Sigma_w freq[w]$
        if $p \geq \theta'$ then
          morphemes$[v] \leftarrow$ morphemes$[v] \cup$ w
  return morphemes

Figure 8: An example subset of the Feature Expression Graphs that are formed from the minimal pairs in Figure 7.

Figure 9: An example of how probabilities are calculated for each feature value pair in a Feature Expression Graph for the feature np-gender.

dard was both noisier and more time consuming than expected. That is, even though the task is based on how a linguistic field worker might collect data, it was more difficult for a human than anticipated. Therefore, we instead produced a list of hypothesized morpheme-feature pairs and had a human trained in linguistics who was also bilingual in Hindi/Urdu-English mark each pair as “Correct,” “Incorrect,” or “Ambiguous.” The results of this evaluation are summarized in Figure 10. The reader may be surprised by how many incorrect hypotheses were generated, given the controlled nature of the Elicitation Corpus. However, there are two important factors to consider. First, features can interact in complex and often unexpected ways. For instance, in English, the only feature difference in minimal pair *Cats yawned* and *A cat yawned* is the number of the actor. However, this causes an interaction with definiteness that would cause the presented algorithms to associate a with the number of nouns even though it is canonically associated with definiteness. Second, the bilingual people translating the Elicitation Corpus are prone to make errors.

Though a fair number of incorrect hypotheses were produced, we were encouraged by the number of correct hypotheses in the output. We also
### Table

<table>
<thead>
<tr>
<th>Judgement</th>
<th>Morpheme-Feature Pairings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>68</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>29</td>
</tr>
<tr>
<td>Incorrect</td>
<td>109</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>206</strong></td>
</tr>
</tbody>
</table>

Figure 10: The results of feature detection. Being a recall-oriented approach, inductive feature detection is geared toward overproduction of morpheme-feature pairings as shown in the number of ambiguous and incorrect pairings.

Note that the words being identified are largely function words and multi-morpheme tokens from which closed-class functional morphemes will be extracted. One might think the counts extracted seem low when compared to the typical MT vocabulary size, but these function words that we extract cover a much larger probability mass of the language than content words.

We are confident that the Morphosyntactic Lexicon Generator designed to operate directly downstream from this process will be sufficiently discriminant to use these morpheme-feature pairings to create a high precision lexicon. However, since this component is, in itself, highly complex, its specifics are beyond the scope of this paper and so we leave it to be discussed in future work.

### 7 Conclusion

We have presented a method for inductive feature detection of an annotated corpus, which determines which feature values have a distinct representation in a target language and what morphemes can be used to express these grammatical meanings. This method exploits the unique properties of an Elicitation Corpus, a resource which is now becoming widely available from the LDC. Finally, we have argued that the output of feature detection is useful for exploiting these linguistic features via a feature-rich grammar for a machine translation system.

### References


Bryant Huang and Kevin Knight. 2006. Relabeling syntax trees to improve syntax-based machine translation quality. In *Proceedings of (NAACL-HLT)*.


