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

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 available 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 systems do not make use of deeper linguistic features including 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 occurrs 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. But where can we get training data that enables us to model these features?

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 accross languages and, thus, which features to include in our feature specification. However, we also know that not every language will display each of these phenomena.

We wish 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 they are each grammaticalized differently. Similarly, English would

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

Figure 1: An example feature specification.

ID	Source Language	Target Language	Lexical Cluster	Feature Structure
s1	He loves her.	El ama a ella.	$\ell 1$	((act (np-gen m) (np-num sg) (np-def +)) (und (np-gen f) (np-num sg) (np-def +)) (c-ten pres))
s2	She loves her.	Ella ama a ella.	$\ell 1$	((act (np-gen f) (np-num sg) (np-def +)) (und (np-gen f) (np-num sg) (np-def +)) (c-ten pres))
s3	He loved her.	El *ama a ella.	$\ell 1$	((act (np-gen m) (np-num sg) (np-def +)) (und (np-gen f) (np-num sg) (np-def +)) (c-ten past))
s4	The boy eats.	El niño come.	$\ell 2$	((act (np-gen m) (np-num sg) (np-def +)) (c-ten pres))
s5	The girl eats.	La niña come.	$\ell 2$	((act (np-gen f) (np-num sg) (np-def +)) (c-ten pres))
s6	A girl eats.	Una niña come.	$\ell 2$	((act (np-gen f) (np-num sg) (np-def -)) (c-ten pres))
s7	The girls eat.	Las niñas comen.	$\ell 2$	((act (np-gen f) (np-num pl) (np-def +)) (c-ten pres))
s8	The girls eat.	Las niñas comen.	$\ell 2$	((act (np-gen f) (np-num dl) (np-def +)) (c-ten pres))
s9	Girls eat.	Unas niñas comen.	$\ell 2$	((act (np-gen f) (np-num pl) (np-def -)) (c-ten pres))

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.

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 Figure 4.

Feature	Value	Candidate Morphemes
np-gen	m	el, niño
np-gen	f	ella, niña
np-gen	n	*unobserved*
np-def	+	el, la, las
np-def	-	una, unas
np-num	sg	el, ella, la, una, come, niño, niña
np-num	dl-pl	las, unas, comen, niñas
c-ten	past-pres	–
c-ten	fut	*unobserved*

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

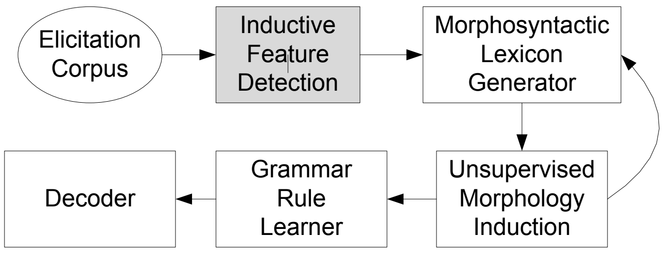
Elicitation Corpus
Inductive Feature Detection
Morphosyntactic Lexicon Generator
Unsupervised Morphology Induction
Grammar Rule Learner
Decoder

Figure 4: An outline of the steps from an input Elicitation Corpus to the application of a morphosyntactially 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. Already, these language packs are available translated into Thai, Bengali, Urdu, Hungarian, Punjabi, Tamil, and Yoruba (See http://projects.ldc.upenn.edu/LCTL/). With structured elicitation corpora already available on a wide scale, there exists plenty of data that can be exploited via feature detection.

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,

ek	**talb**	alm	arshad	jo	mchhlyoN	ke liye	pani	maiN	aata	phink	**raha**	**tha** . . .
a.SG	**student**	named	Irshad	who	fish	for	water	in	flour	throw	**PROG.SG.M**	**be.**PAST.SG.M

"A student named Irshad who was throwing flour in the water for the fish . . . "

Figure 5: A glossed e from parallel text in LDC's Urdu-English LCTL language pack showing subject-verb agreement being separated by 12 words.

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 (Poornima 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 [1].

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 specifiction 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 previously stated, feature detection uses an Elicitation Corpus (see Figure 2), a corpus that has been carefully constructed to provide a large number of

[1] If the reader is not familiar with Unification Grammars, we recommend Kaplan (1995)

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. 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. 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.

Also, 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*.

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 strucutres* 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 Morphosyntatic 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 wilcards 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 $m \in$ minimalPairs do
    for all sentence $s \in$ m do
      f ← wildcardFeature(s, m)
      v ← featureValue(s, f)
      c ← featureContext(m)
      $\ell$ ← lexCluster(s)
      T[$f, m, c, \ell, v$] ← T[$f, m, c, \ell, v$]$\cup$ s
  return T
```

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*. We define a feature expression graph (FEG) for a feature f be 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

ID	Set Members	Feature	Feature Context	Feature Structure
m1	{s1, s2}	np-gen	((act))	((act (np-gen *) (np-num sg) (np-def +)) (und (np-gen f) (np-num sg) (np-def +)) (c-ten pres))
m2	{s1, s3}	np-ten	()	((act (np-gen m) (np-num sg) (np-def +)) (und (np-gen f) (np-num sg) (np-def +)) (c-ten *))
m3	{s4, s5, s7, s8}	np-gen	((act))	((act (np-gen *) (np-num sg) (np-def +)) (c-ten pres))
m4	{s5, s7, s8}	np-num	((act))	((act (np-gen f) (np-num *) (np-def +)) (c-ten pres))
m5	{s6, s9}	np-num	((act))	((act (np-gen f) (np-num *) (np-def -)) (c-ten pres))
m6	{s5, s6}	np-def	((act))	((act (np-gen f) (np-num sg) (np-def *)) (c-ten pres))
m7	{s7, s9}	np-def	((act))	((act (np-gen f) (np-num pl) (np-def *)) (c-ten pres))
etc.				

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

Feature	Min. Pair	Feat. Context	Lex. Cluster	Feat. Value.	Sentence
np-gen	m1	((act))	$\ell 1$	m	s1
np-gen	m1	((act))	$\ell 1$	f	s2
np-ten	m2	()	$\ell 1$	pres	s1
np-ten	m2	()	$\ell 1$	past	s3
np-num	m4	((act))	$\ell 2$	sg	s5
np-num	m4	((act))	$\ell 2$	pl	s7
np-num	m4	((act))	$\ell 2$	dl	s8
np-num	m5	((act))	$\ell 2$	sg	s6
np-num	m5	((act))	$\ell 2$	pl	s9
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.

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.

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.
  for all $s_i, s_j \in$ T s.t. $(m_i, c_i, \ell_i) = (m_j, c_j, \ell_j)$ do
    $(v_i, v_j) \leftarrow$ (featureValue($s_i$), featureValue($s_j$))
    if tgt($s_i$) = tgt($s_j$) then
      arcs$[v_i, v_j] \leftarrow$ arcs$[v_i, v_j] \cup (s_i, s_j, m,$EQ)
    else
      arcs$[v_i, v_j] \leftarrow$ arcs$[v_i, v_j] \cup (s_i, s_j, m,$NEQ)
  return arcs
```

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 θ. 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:

```
np-gen: m, f
np-num: s, pl/dl
np-def: +, -
c-ten: past, pres
```

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.

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 ∈ F do
    for all vertex ∈ feature do
      for all arc ∈ 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 ∈ freq do
            freq$[w]$++
        for all w ∈ freq do
          p = freq$[w]$ / $\Sigma_w$ freq$[w]$
          if $p \geq \theta'$ then
            morphemes$[v] \leftarrow$ morphemes$[v] \cup$ w
  return morphemes
```

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 manually chosen to be 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 standard was both noisier and more time consuming than expected. That is, even though this task is

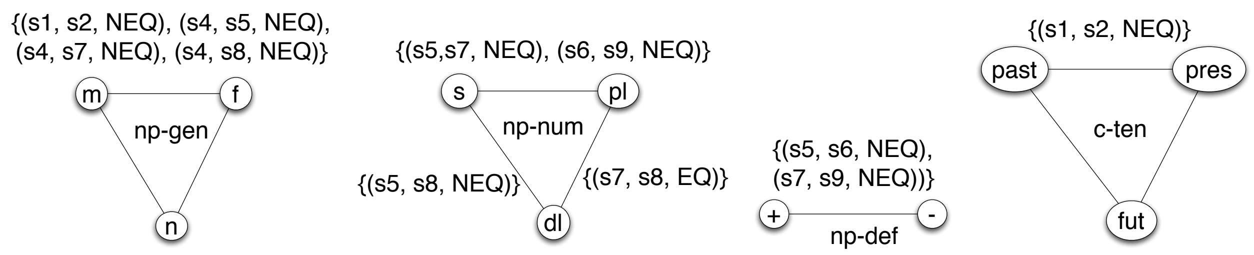

Figure 8: An example subset of the Feature Expression Graphs that are formed from the minimal pairs in Figure 7.

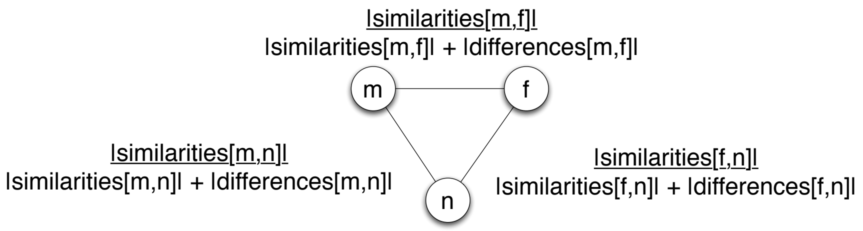
|similarities[m,f]|
|similarities[m,f]| + |differences[m,f]|
m
f
|similarities[m,n]|
|similarities[m,n]| + |differences[m,n]|
n
|similarities[f,n]|
|similarities[f,n]| + |differences[f,n]|

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

based on how a linguistic field worker might collect data, this task was much more difficult for a human than originally anticipated. Therefore, we instead produced a list of hypothesized morpheme-feature pairs and then 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. Though a fair number of incorrect hypotheses were produced, we were encouraged by the number of correct hypotheses in the output. We also 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.

Judgement	**Morpheme-Feature Pairings**
Correct	68
Ambiguous	29
Incorrect	109
TOTAL	206

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.

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

H. Bonneau-Maynard, A. Allauzen, D. Déchelotte, and H. Schwenk. 2007. Combining morphosyntactic enriched representation with n-best reranking in statistical translation. In *Proceedings of the Workshop on Structure and Syntax in Statistical Translation (SSST) at NAACL-HLT*.

Jaime Carbonell, Kathrina Probst, Erik Peterson, Christian Monson, Alon Lavie, Ralf Brown, and Lori Levin. 2002. Automatic rule learning for resource limited MT. In *Association for Machine Translation in the Americas (AMTA)*, October.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Association for Computational Linguistics (ACL)*.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics ACL*.

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

Ronald Kaplan. 1995. The formal architecture of lexical functional grammar. In Mary Dalrymple, Ronald Kaplan, J. Maxwell, and A. Zaenen, editors, *Formal Issues in Lexical Functional Grammar*. CSLI Publications.

Paul Kay. 2002. An informal sketch of a formal architecture for construction grammar. In *Grammars*.

Phillipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Percy Liang, Alexandre Bouchard-Cote, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney.

Christian Monson, Jaime Carbonell, Alon Lavie, and Lori Levin. 2007. Paramor: Minimally supervised induction of paradigm structure and morphological analysis. In *Proceedings of the 9th ACL SIGMORPH*.

Shakthi Poornima and Jean-Pierre Koenig. 2008. Reverse complex predicates in Hindi. In *Proceedings of the 24th Northwest Linguistic Conference*.