Controlled Ascent: Imbuing Statistical MT with Linguistic Knowledge

William D. Lewis and Chris Quirk
Microsoft Research
One Microsoft Way
Redmond, WA 98052
{wilewis,chrisq}@microsoft.com

Abstract

We explore the intersection of rule-based and statistical approaches in machine translation, with a particular focus on past and current work here at Microsoft Research. Until about ten years ago, the only machine translation systems worth using were rule-based and linguistically-informed. Along came statistical approaches, which use large corpora to directly guide translations toward expressions people would actually say. Rather than making local decisions when writing and conditioning rules, goodness of translation was modeled numerically and free parameters were selected to optimize that goodness. This led to huge improvements in translation quality as more and more data was consumed. By necessity, the pendulum is swinging towards the inclusion of linguistic features in MT systems. We describe some of our statistical and non-statistical attempts to incorporate linguistic insights into machine translation systems, showing what is currently working well, and what isn't. We also look at trade-offs in using linguistic knowledge ("rules") in pre- or post-processing by language pair, with a particular eye on the return on investment as training data increases in size.

1 Introduction

Machine translation has undergone several paradigm shifts since its original conception. Early work considered the problem as cryptography, imagining that a word replacement cipher could find the word correspondences between two languages. Clearly Weaver was decades ahead of his time in terms of both computational power and availability of data: only now is this approach gaining some traction (Knight, 2013)[1] At the time, however, this direction did not appear promising, and work turned toward rule-based approaches.

Effective translation needs to handle a broad range of phenomena. Word substitution ciphers may address lexical selection, but there are many additional complexities: morphological normalization in the source language, morphological inflection in the target language, word order differences, and sentence structure differences, to name a few. Many of these could be captured, at least to a first degree of approximation, by rule-based approaches. A single rule might capture the fact that English word order is predominantly SVO and Japanese word order is predominantly SOV. While many exceptions exist, such rules handle many of the largest differences between languages rather effectively. Therefore, rule-based systems that did a reasonable job of addressing morphological and syntactic differences between source and target dominated the marketplace for decades.

With the broader usage of computers, greater amounts of electronic data became available to systems. Example-based machine translation systems, which learn corpus-specific translations based on data, began to show substantial improvements in the core problem of lexical selection. This task was always quite difficult for rule-based approaches: finding the correct translation in context requires a large amount of knowledge. In practice, nearby words are effective disambiguators once a large amount of data has been captured.

Phrasal statistical machine translation systems formalized many of the intuitions in example-based machine translation approaches, replacing heuristic selection functions with robust statistical estimators. Effective search techniques developed originally for speech recognition were strong starting influences in the complicated realm of MT decoding. Finally, large quantities of parallel data and even larger quantities of monolingual data allowed such phrasal methods to shine even in broad domain translation.

Translations were still far from perfect, though. Phrasal systems capture local context and local reordering well, but struggle with global reordering. Over the past decade, statistical machine translation has begun to be influenced by linguistic information once again. Syntactic models have shown some of the most compelling gains. Many systems leverage the syntactic structure of either the

[1] For the original 1949 *Translation* memorandum by Weaver see (Weaver, 1955).

Proceedings of the Second Workshop on Hybrid Approaches to Translation, pages 51–66,
Sofia, Bulgaria, August 8, 2013. ©2013 Association for Computational Linguistics

source or the target sentences to make better decisions about reordering and lexical selection.

Our machine translation group has been an active participant in many of these latest developments. The first MSR MT system used deep linguistic features, often with great positive effect. Inspired by the successes and failures of this system, we invested heavily in syntax-based SMT. However, our current statistical systems are still linguistically impoverished in comparison.

This paper attempts to document important lessons learned, highlight current best practices, and identify promising future directions for improving machine translation. A brief review of our earlier generation of machine translation technology sets the stage; this older system remains relevant given renewed interest in semantics (*e.g.*, http://amr.isi.edu/). Next we describe some of our statistical and non-statistical attempts to incorporate linguistic insights into machine translation systems, showing what is currently working well, and what is not. We also look at trade-offs in using linguistic knowledge ("rules") in pre- or post-processing by language pair, with a particular eye on the return on investment as training data increases in size. Systems built on different architectures, particularly those incorporating some linguistic information, may have different learning curves on data. The advent of social media and big data presents new challenges; we review some effective research in this area. We conclude by exploring promising directions for improving translation quality, especially focusing on areas that stand to benefit from linguistic information.

2 Logical Form Translation

Machine translation research at Microsoft Research began in 1999. Analysis components had been developed to parse surface sentences into deep *logical forms*: predicate-argument structures that normalized away many morphological and syntactic differences. This deep representation was originally intended for information mining and question answering, allowing facts to reinforce one another, and simplifying question and answer matching. These same normalizations helped make information more consistent across languages: machine translation was a clear potential application. Consider the deep representations of the sentence pairs in Figure 1: many of the surface differences, such as word order and morphological inflection, are normalized away, potentially easing the translation process.

```
To please reviewers is difficult.
be1 (+Pres +Proposition)
 ╲Tobj───difficult1 (+Proposition +F0)
          ╲Tsub───please1 (+Inf +Psych)
                   ╲Tsub───_X1
                   ╲Tind───reviewer1 (+Plur)
Reviewers are difficult to please.
be1 (+Pres +Proposition)
 ╲Tobj───difficult1 (+Proposition +B3)
          ╲Tsub───please1 (+Inf +Psych)
                   ╲Tsub───_X1
                   ╲Tind───reviewer1 (+Plur)
Pleasing reviewers is difficult.
be1 (+Pres +Proposition)
 ╲Tobj───difficult1 (+Proposition +F0)
          ╲Tsub───please1 (+Verbnoun +Psych)
                   ╲Tsub───_X1
                   ╲Tind───reviewer1 (+Plur)
```

Figure 1: Example logical forms for three distinct inputs, demonstrating how differences in syntactic structure may be normalized away. In each case, the logical form is a graph of nodes such as "be" and "difficult", and relations such as "Tobj" (typical object) and "Tsub" (typical subject). In addition, nodes are marked with binary features called bits, prefixed with a + symbol in the notation, that capture unstructured pieces of information such as tense and number.

Substantial differences remained, however. Many words and phrases have non-compositional contextually-influenced translations. Commercial systems of the time relied on complex, hand-curated dictionaries to make this mapping. Yet example-based and statistical systems had already begun to show promise, especially in the case of domain-specific translations. Microsoft in particular had large internal demand for "technical" translations. With increasing language coverage and continuing updates to product documentation and support articles came increasing translation costs. Producing translations tailored to this domain would have been an expensive task for a rule-based system; a corpus-based approach was pursed.

This was truly a hybrid system. Source and target language surface sentences were parsed into deep logical forms using rule-based analyzers.[2]

[2] These parsers were developed with a strong focus on corpora, though. George Heidorn, Karen Jensen, and the NLP research group developed a toolchain for quickly parsing a large bank of test sentences and comparing against the last best result. The improvements and regressions resulting from a change to the grammar could be manually evaluated, and the changes refined until the end result. The end result was a

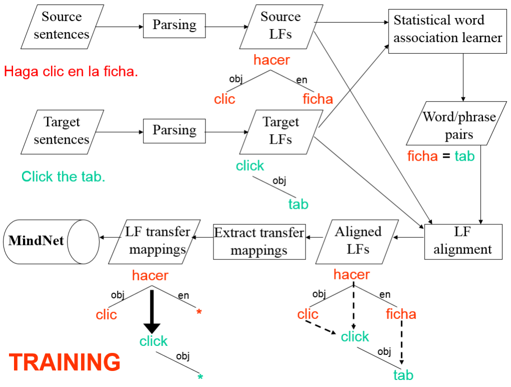


Figure 2: The process of learning translation information from parallel data in the LF system.

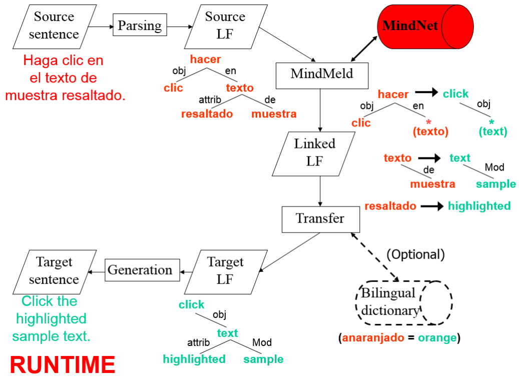


Figure 3: The process of translating a new sentence in the LF system.

Likewise a rule-based target language generation component could find a surface realization of a deep logical form. However, the mapping from source language logical form fragments to target language logical form fragments was learned from parallel data.

2.1 Details of the LF-based system

Training started with a parallel corpus. First, the source and target language sentences were parsed. Then the logical forms of the source and target were aligned (Menezes and Richardson, 2001). These aligned logical forms were partitioned into minimal non-compositional units, each consisting of some non-empty subset of the source and target language nodes and relations. Much like in example-based or phrasal systems, both minimal and composed versions of these units were then stored as possible translations. A schematic of the this data flow is presented in Figure 2.

At runtime, an input sentence was first parsed into a logical form. Units whose source sides matched the logical form were gathered. A heuristic search found a set of fragments that: (a) covered every input node at least once, and (b) were consistent in their translation selections. If some node or relation was not uncovered, it was copied from source to target. The resulting target language logical form was then fed into a generation component, which produced the final string. A schematic diagram is presented in Figure 3.

This overview sweeps many fine details under the rug. Many morphological and syntactic distinctions were represented as binary features ("bits") in the LF; mapping bits was difficult. The logical form was a graph rather than a tree – in "John ate and drank", *John* is the DSUB (deep subject) of both *eat* and *drink* – which led to complications in transferring structure. Many such complications were often handled through rules; these rules grew more complex over time. Corpus-based approaches efficiently learned many non-compositional and domain specific issues.

2.2 Results and lessons learned

The system was quite successful at the time. MSR used human evaluation heavily, performing both absolute and relative quality evaluations. In the absolute case, human judges gave each translation a score between 1 (terrible translation) and 4 (perfect). For relative evaluations, judges were presented with two translations in randomized order, and were asked whether they preferred system A, system B, or neither. In its training domain, the LF-based system was able to show substantial improvements over rule-based systems that dominated the market at the time.

Much of these gains were due to domain- and context-sensitivity of the system. Consider the Spanish verb "activar". A fair gloss into English is "activate", but the most appropriate translation in context varies ("signal", "flag", etc.). The example-based approach was able to capture those contexts very effectively, leading to automatic domain customization given only translation memories. This was a huge improvement over rule-based systems of the time.

During this same era, however, statistical approaches (Och and Ney, 2004) were showing great promise. Therefore, we ran a comparison between the LF-based system and a statistical system

data driven but not statistical approach to parser development.

(a) Effecitve LF translation. Note how the LF system is able to translate "se lleveban a cabo" even though that particular surface form was not present in the training data.

SRC: La tabla muestra además dónde se llevaban a cabo esas tareas en Windows NT versión 4.0.
REF: The table also shows where these tasks were performed in Windows NT version 4.0.
LF: The table shows where, in addition, those tasks were conducted on Windows NT version 4.0.
STAT: The table also shows where llevaban to Windows NT version 4.0.

(b) Parsing errors may degrade translation quality; the parser interprted '/' as coordination.

SRC: La sintaxis del operador / tiene las siguientes partes:
REF: The / operator syntax has these parts:
LF: The operator syntax it has the parts:
STAT: The / operator syntax has these parts:

(c) Graph-like structures for situations such as coordination are difficult to transfer (see the parenthesized group in particular); selecting the correct form at generation time is difficult in the absence of a target language model.

SRC: Debe ser una consulta de selección (no una consulta de tabla de referencias cruzadas ni una consulta de acción).
REF: Must be a select query (not a crosstab query or action query).
LF: You must not be a select query neither not a query in table in cross-references nor not an action query.
STAT: Must be a select query (not a crosstab query or an action query).

Figure 4: Example source Spanish sentences, English reference translations of those sentences, translations from the LF system, and translations from a statistical translation system without linguistic features.

without linguistic information. Both systems were trained and tuned on the same data, and translated the same unseen test set. The linguistic system had the additional knowledge sources at its disposal: morphological, lexical, syntactic, and semantic information. Regardless, the systems performed nearly equally well on average. Each had distinct strengths and weaknesses, though.

Often the success or failure of the LF-system was tied to the accuracy of its deep analysis. When these representations were accurate, they could lead to effective generalizations and better translations of rare phenomena. Since surface words were lemmatized and syntactic differences normalized, unseen surface forms could still be translated as long as their lemma was known (see Figure 4(a)). Yet mistakes in identifying the correct logical form could lead to major translation errors, as in Figure 4(b).

Likewise the lack of statistics in the components could cause problems. Statistical approaches found great benefits from the target language model. Using a rule-based generation component made it difficult to leverage a target language model. Often, even if a particular translation was presented tens, hundreds, or thousands of times in the data, the LF-based system could not produce it because the rule-based generation component would not propose the common surface form, as in Figure 4(c).

We drew several lessons from this system when developing our next generation of machine translation systems. It was clear to us that syntactic representations can help translation, especially in reordering and lexical selection: appropriate representations allows better generalization. However, over-generalization can lead to translation error, as can parsing errors.

3 The Next Generation MSR MT Systems

Research in machine translation at Microsoft has been strongly influenced by this prior experience with the LF system. First we must notice that there is a huge space of possible translations. Consider human reference translations: unless tied to a specific domain or area, they seldom agree completely on lexical selection and word order. If our system is to produce reasonable output, it should consider a broad range of translation options, preferring outputs most similar to language used by humans. Why do we say "order of magnitude" rather than "magnitude order", or "master of ceremonies" rather than "ceremonies master"? Many choices in language are fundamentally arbitrary, but we need to conform to those arbitrary decisions if we are to produce fluent and understandable output. Second, while there is leverage to be gained from deep features, seldom do we have a component that identifies these features with perfect accuracy. In practice it seems that the error rate increases as the depth of component analysis increases. Finally, we need a representation of "good translations" that is understandable by a computer. When forced to choose between two translations, the system needs to make a choice: an ordering.

Therefore, our data-driven systems crucially rely on several components. First, we must efficiently search a broad range of translations. Second, we must rank according to both our linguistic intuitions and the patterns that emerge from data.

We use a number of different systems based on the availability of linguistic resources. So-called *phrasal* statistic machine translation systems, which model translations using no more than sequences of contiguous words, perform surprisingly well and require nothing but tokenization in both languages. In language pairs for which we have a source language parser, a parse of the input sentence is used to guide reordering and help select relevant non-contiguous units; this is the *treelet* system (Quirk and Menezes, 2006). Regardless of which system we use, however, target language models score the fluency of the output, and have a huge positive impact on translation quality.

We are interested in means of incorporating linguistic intuition deeper into such a system. As in the case of the treelet system, this may define the broad structure of the system. However, there are also more accessible ways of influencing existing systems. For instance, linguists may author features that identify promising or problematic translations. We describe one such attempt in the following system.

3.1 Like and DontLike

Even in our linguistically-informed treelet system (Quirk and Menezes, 2006), which uses syntax in its translation system, many of the individual mappings are clearly bad, at least to a human. When working with linguistic experts, one gut response is to write rules that inspect the translation mappings and discard those translation mappings that appear dangerous. Perhaps they seem to delete a verb, perhaps they use a speculative reordering rule – something makes them look bad to a linguist. However, even if we are successful in removing a poor translation choice, the remaining possibilities may be even worse – or perhaps no translation whatsoever remains.

Instead, we can soften this notion. Imagine that a linguist is able to say that this mapping is not preferred because of some property. Likewise, a skilled linguist might be able to identify mappings that look particularly promising, and prefer those mappings to others; see Figure 5 for an example.

This begs the question: how much should we weight such influence? Our answer is a corpus driven one. Each of these linguistic preferences should be noted, and the weight of these preferences should be tuned with all others to optimize the goodness of translation. Already our statistical system has a number of signals that attempt to gauge translation quality: the translation models attempt to capture fidelity of translation; language models focus on fluency; etc. We use techniques such as MERT (Och, 2003) and PRO (Hopkins and May, 2011) to tune the relative weight of these signals. Why not tune indicators from linguists in the same manner?

When our linguists mark a mapping as +Like or +DontLike, we track that throughout the search. Each final translation incorporates a count of Like mappings and a count of DontLike mappings, just as it accumulates a language model score, translation model scores, word penalties, and so on. These weights are tuned to optimize some approximate evaluation metric. In Figure 6, the weight of Like and DontLike is shown for a number of systems, demonstrating how optimization may be used to tune the effect of hand-written rules. Removing these features degrades the performance of an MT system by at least 0.5 BLEU points, though the degradations are often even more visible to humans.

This mechanism has been used to capture a number of effects in translation commonly missed by statistical methods. It is crucial yet challenging to maintain negation during translation, especially in language pairs where negation is expressed differently: some languages use a free morpheme (Chinese tends to have a separate word), others use a bound morpheme (English may use prefixes), others require two separated morphemes (French has negation agreement); getting any of these wrong can lead to poor translations. Rules that look at potentially distant words can help screen away negation errors. Likewise rules can help ensure that meaning is preserved, by preventing main verbs mapping to punctuation, or screen-

```
// don't allow verb to be lost
if (forany(NodeList(rMapping),[Cat=="Verb" & ^Aux(SynNode(InputNode))])) {
    list {segrec} bad_target=sublist(keeplist,
        [forall(NodeList,[pure_punk(Lemma) | coord_conjunction(foreign_language,Lemma)])]);
    if (bad_target) {
        segrec rec;
        foreach (rec; bad_target) {
            +DontLike(rec);
        }
    }
}
```

Figure 5: An example rule for marking mappings as “DontLike”. In this case, the rule searches for source verbs that are not auxiliaries and that are translated into lemmas or punctuation. Such translations are marked as DontLike.

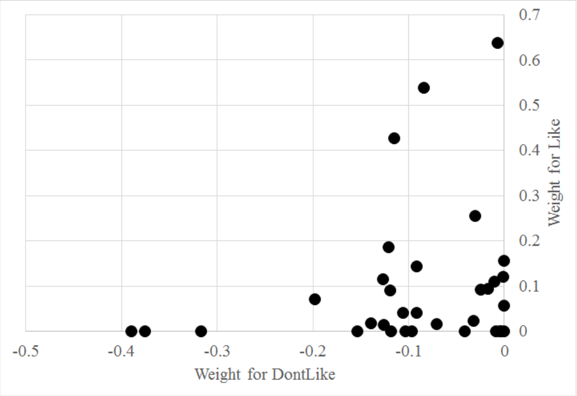


Figure 6: A plot of the weights +Like mapping count and +DontLike mapping count weights across language pairs. Generally Like is assigned a positive weight (sometimes quite positive), and DontLike is assigned a negative weight. In our system, weights are L1 normalized (the sum of the absolute values of the weights is equal to one), so feature weights greater than 0.1 are very influential.

ing out mappings that seem unlikely, especially when those mappings involve unusual tokens.

These two features are a rather coarse means of introducing linguistic feedback. As our parameter estimation techniques scale to larger features more effectively, we are considering using finer-grained feedback from linguists to say not only that they like or don't like a particular mapping, but why. The relative impact of each type of feedback can be weighted: perhaps it is critical to preserve verbs, but not so important to handle definiteness. Given recent successes in scaling parameter estimation to larger and larger values, this area shows great promise.

3.2 Linguistic component accuracy

Another crucial issue is the quality of the linguistic components. We would certainly hope that better quality of linguistic analysis should lead to better quality translations. Indeed, in certain circumstances it appears that this correlation holds.

In the case of the treelet system, we hope to derive benefit from linguistic features via a dependency tree. To investigate the impact of the parse quality, we can degrade a Treebank-trained parser by limiting the amount of training data made available. As this decreases, the parser quality should degrade. If we hold all other information in the MT system fixed (parallel and monolingual training data, training regimen, etc.), then all differences should be due to the changes in parse quality. Table 1 presents the results of an experiment of this form (Quirk and Corston-Oliver, 2006). As the amount of training data increase, we see a substantial increase in parse quality.

Another way to mitigate parser error is to maintain syntactic ambiguity through the translation process. For syntax directed translation systems, this can be achieved by translating forests rather than single trees, ideally including the score of

System	English-German	English-Japanese
Phrasal	31.7	32.9
Right branching	31.4	28.0
250 instances	32.8	34.1
2,500 instances	33.0	34.6
25,000 instances	33.7	35.7
39,082 instances	33.8	36.0

Table 1: Comparison of BLEU scores as linguistic information is varied. A phrasal system provides a baseline free of linguistic information. Next we consider a treelet system with a very weak baseline: a right branching tree is always proposed. This baseline is much worse than a simple phrasal system. The final four rows evaluate the impact of a parser trained on increasing amounts of sentences from the English Penn Treebank. Even with a tiny amount of training data, the system gets some benefit from syntactic information, and the returns appear to increase with more training data.

parse as part of the translation derivation. In unpublished results, we found that this made a substantial improvement in translation quality; the effect was corroborated in other syntax directed translation systems (Mi et al., 2008). Alternatively, allowing a neighborhood of trees similar to some predicted tree can handle ambiguity even when the original parser does not maintain a forest. This also allows translation to handle phenomena that are systematically mis-parsed, as well as cases where the parser specification is not ideal for the translation task. Recent work in this area has show substantial improvements (Zhang et al., 2011).

4 Evaluation

4.1 Fact or Fiction: BLEU is Biased Against Rule-Based or Linguistically-Informed Systems?

It has generally been accepted as common wisdom that BLEU favors statistical MT systems and disfavors those that are linguistically informed or rule-based. Surprisingly, the literature on the topic is rather sparse, with some notable exceptions (Riezler and Maxwell, 2005; Farrús et al., 2012; Carpuat and Simard, 2012). We too have made this assumption, and had a few years ago coined the term *treelet penalty* to indicate the degree by which BLEU favored our phrasal systems over our treelet systems. We had noted on a few occasions that treelet systems had lower BLEU scores than our phrasal systems over the same data (the "penalty"), but when compared against one another in human evaluation, there was little difference, or often, treelet was favored. A notable case was on German-English, where we noted a three-point difference in BLEU between equivalent treelet and phrasal systems (favoring phrasal), and a ship/no-ship decision was dependent on the resulting human eval. The general consensus of the team was that the phrasal system was markedly better, based on the BLEU result, and treelet system should be pulled. However, after a human eval was conducted, we discovered that the treelet system was significantly better than the phrasal. From that point forward, we talked about the *treelet penalty* for German being three points, a "fact" that has lived in the lore of our team ever since.

What was really missing, however, was systematic experimental evidence showing the differences between treelet and phrasal systems. We talked about the treelet penalty as a given, but there was slow rumble of counter evidence suggesting that maybe the assumptions behind the "penalty" were actually unfounded, or minimally, misinformed.

One piece of evidence was from experiments done by Xiaodong He and an intern that showed an interaction in quality differences between treelet and phrasal gated by the length of the sentence. Xiaodong was able to show that phrasal systems tended to do better on longer sentences and treelet on shorter: for Spanish-English, he showed a difference in BLEU of 1.29 on "short" content on a general domain test set, and 1.77 for short content on newswire content (the NIST08 test set). The BLEU difference diminished as the length of the content increased, until there was very little difference (less than 1/2 point) for longer content.[3] An interaction between decoder type and sentence length means that there might also be an interac-

[3] These results were not published, but were provided to the authors in a personal conversation with Xiaodong. In a related paper (He et al., 2008), He and colleagues showed significant improvements in BLEU on a system combination system, but no diffs in human eval. Upon analysis, the researchers were able to show that the biggest benefit to BLEU was in short content, but the same preference was not exhibited on the same content by the human evaluators. In other words, the improvements observed in the short content that BLEU favored had little impact on the overall impressions of the human evaluators.

tion between decoder type and test set, especially if particular test sets contain a lot of long-ish sentences, *e.g.,* WMT and Europarl). To the contrary, most IT text, which is quite common in Microsoft-specific localization content, tends to be shorter.

The other was based on general impressions between treelet and phrasal systems. Because treelet systems are informed by dependency parses built over the source sentences (a parse can help constrain a search space of possible translations, and prune undesirable mappings *e.g.,* constrain to nominal types when the source is a noun), and, as noted earlier, because the parses allow linguists to pre- or post-process content based on observations in the parse, we have tended to see more "fluent" output in treelet than phrasal. However, as the sizes of data have grown steadily over the years, the quality of translations in our phrasal systems have grown proportionally with the increase in data. The question arose: is there also an interaction between the size of our training data and decoder type? In effect, does the quality of phrasal systems catch-up to the quality of treelet systems when trained over very large sets of data?

4.2 Treelet Penalty Experiments

We ran a set of experiments to measure the differences between treelet and phrasal systems over varying sizes of data, in order to measure the size of the treelet penalty and its interaction with training data size. Our assumption was that a such a penalty existed, and that the penalty decreased as training data size increased, perhaps converging on zero for very large systems. Likewise, we wanted to test the interaction between decoder type and sentence length.

We chose two languages to run these experiments on, Spanish and German, which we ran in both directions, that is, English-to-target (EX) and target-to-English (XE). We chose Spanish and German for several reasons, first among them being that we have high-quality parsers for both languages, as we do for English. Further, we have done significant development work on pre- and post-processing for both languages over the past several years. Both of these facts combined meant that the treelet systems stood a real chance of being strong contenders in the experiments against the equivalent phrasal systems. Further, although the languages are typologically close neighbors of English, the word order differences and high distortion rates from English to or from German might favor a parser-based approach.

We had four baseline systems that were built over very large sets of data. For Spanish ⇄ English, the baseline systems were trained on over 22M sentence pairs; for German ⇄ English, the baseline systems were trained on over 36M sentence pairs.[4] We then created five samples of the baseline data for each language pair, consisting of 100K, 500K, 1M, 2M, and 5M sentence pairs (the same samples were used for both EX and XE for the respective pairs). We then trained both treelet and phrasal systems in both directions (EX and XE) over each sample of data. Language models were trained on all systems over the target-side data.

For dev data, we used development data from the 2010 WMT competition (Callison-Burch et al., 2010), and we used MERT (Och, 2003) to tune each system. We tested each system against three different test sets: two were from the WMT competitions of 2009 and 2010, and the other was one locally constructed from 5000 sentences of content translated by users of our production service (http://bing.com/translator), which we subsequently had manually translated into the target languages. The former two test sets are somewhat news focused; the latter is a random sample of miscellaneous translations, and is more generally focused.

The results of the experiments are shown in Tables 2 and 3, with the relevant graphs in Figures 9 - 10. The reader will note that in *all* cases—Spanish and German, EX and XE—the treelet systems scored higher than the related phrasal systems. This result surprised us, since we thought that treelet systems would score *less* than phrasal systems, especially at lower data sizes. That said, in the Spanish systems, there is a clear convergence as data sizes increased: on the WMT09 test set for English-Spanish, for instance, the diff starts at 1.46 BLEU (treelet minus phrasal) for the 100K sentence system, with a steady convergence to near zero (0.12) for the full-data baseline. The other test sets show the same steady convergence, although they do not approach zero quite as closely. (One might ask whether they would converge to zero with more training data.) The

[4] A sizable portion of the data for each were scraped from the Web, but there were other sources used as well, such as Europarl, data from TAUS, MS internal localization data, UN content, WMT news content, etc.

other direction is even more dramatic: on all test sets the diffs converge on negative values, indicating that phrasal systems surpass the quality of the associated treelet systems at the largest data points. This is a nice result since it shows, at least in the case of Spanish, that there is an interaction between decoder type and the amount of data: treelet clearly does better at lower data amounts, but phrasal catches up with, and can even pass, the quality of equivalent treelet given sufficient data. With larger data, phrasal may, in fact, be favored over treelet.

The German systems do not tell quite as nice a story. While it is still true that treelet has higher BLEU scores than phrasal throughout, and that systems trained using both decoders improve in quality as more data is added (and the trajectory is similar), there is no observable convergence as data size increases. For German, then, we can only say that more data helps either decoder, but we cannot say that phrasal benefits from larger data more than treelet. Why the difference between Spanish and German? We suspect there may be an interaction with the parsers, in that two separate teams developed them. Thus, it could be the fact that the strength of the respective parsers affected how "linguistically informed" particular systems are. There could also be an interaction with the number of word types vs. tokens in the German data—given German's rampant compounding—which increases data sparsity, dampening effects until much larger amounts of data are used. We are still in the process of running additional experiments to see if there are observable effects in German with much larger data sizes, or at least, to determine why German does not show the same effects as Spanish.

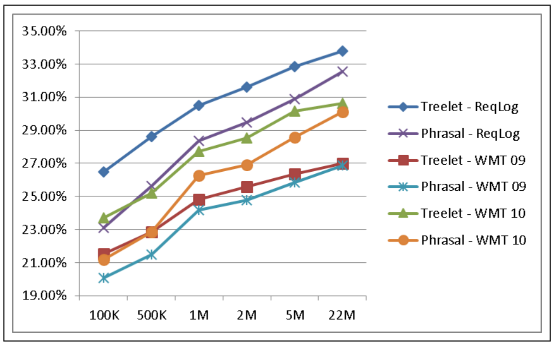


Figure 7: English-Spanish BLEU graph across different data sizes, Treelet vs. Phrasal.

Since human evaluation is the gold standard we

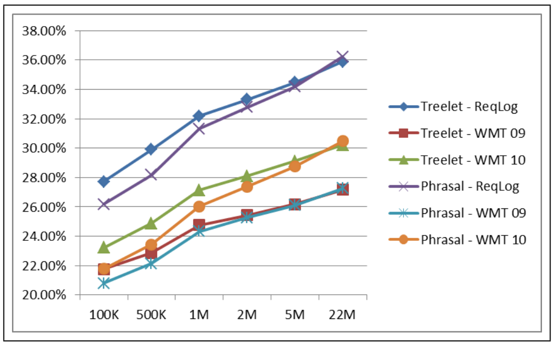


Figure 8: Spanish-English BLEU graph across different data sizes, Treelet vs. Phrasal.

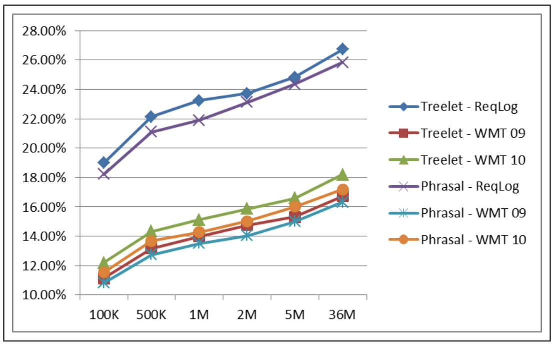


Figure 9: English-German BLEU graph across different data sizes, Treelet vs. Phrasal.

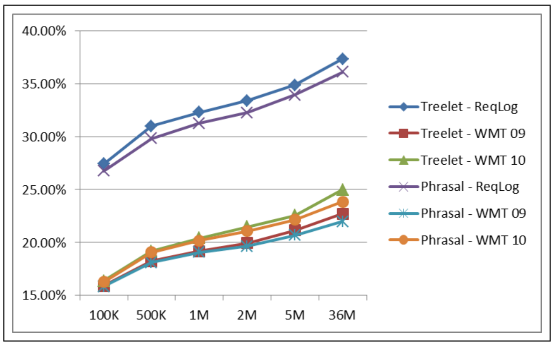


Figure 10: German-English BLEU graph across different data sizes, Treelet vs. Phrasal.

EX		Treelet			Phrasal			Diff - T-P	
	Req Log	WMT 2009	WMT 2010	Req Log	WMT 2009	WMT 2010	Req Log	WMT 2009	WMT 2010
100K	26.49	21.52	23.69	23.10	20.06	21.19	3.39	1.46	2.50
500K	28.61	22.85	25.20	25.64	21.47	22.86	2.97	1.38	2.34
1M	30.52	24.82	27.74	28.36	24.17	26.28	2.16	0.65	1.46
2M	31.61	25.59	28.54	29.48	24.76	26.91	2.13	0.83	1.63
5M	32.86	26.37	30.14	30.89	25.84	28.56	1.97	0.53	1.58
22M	33.80	27.01	30.61	32.55	26.89	30.12	1.25	0.12	0.49
XE									
100K	27.72	21.76	23.21	26.18	20.80	21.78	1.54	0.96	1.43
500K	29.89	22.86	24.89	28.16	22.15	23.44	1.73	0.71	1.45
1M	32.18	24.76	27.14	31.32	24.32	26.02	0.86	0.44	1.12
2M	33.31	25.44	28.09	32.77	25.26	27.38	0.54	0.18	0.71
5M	34.47	26.17	29.10	34.18	26.10	28.74	0.29	0.07	0.36
22M	35.88	27.16	30.20	36.21	27.26	30.48	-0.33	-0.10	-0.28

Table 2: BLEU Score results for the Spanish Treelet Penalty experiments

EX		Treelet			Phrasal			Diff (T-P)	
	Req Log	WMT 2009	WMT 2010	Req Log	WMT 2009	WMT 2010	Req Log	WMT 2009	WMT 2010
100K	18.98	11.13	12.19	18.22	10.81	11.53	0.76	0.32	0.66
500K	22.13	13.18	14.33	21.09	12.74	13.68	1.04	0.44	0.65
1M	23.23	13.98	15.12	21.89	13.51	14.27	1.34	0.47	0.85
2M	23.72	14.77	15.87	23.11	14.04	15.03	0.61	0.73	0.84
5M	24.82	15.31	16.58	24.35	15.00	16.01	0.47	0.31	0.57
36M	26.72	16.72	18.20	25.83	16.33	17.18	0.89	0.39	1.02
XE									
100K	27.42	15.91	16.37	26.75	15.83	16.28	0.67	0.08	0.09
500K	30.98	18.25	19.16	29.80	18.11	19.09	1.18	0.14	0.07
1M	32.30	19.16	20.40	31.26	19.06	20.18	1.04	0.10	0.22
2M	33.40	19.95	21.48	32.25	19.65	21.06	1.15	0.30	0.42
5M	34.86	21.14	22.55	33.91	20.67	22.13	0.95	0.47	0.42
36M	37.31	22.72	24.97	36.08	21.99	23.85	1.23	0.73	1.12

Table 3: BLEU Score results for the German Treelet Penalty experiments

seek to achieve with our quality measures, and since BLEU is only weakly correlated with human eval (Coughlin, 2003), we ran human evals against both the English-Spanish and English-German output. Performing human evaluation gives us two additional perspectives on the data: (1) do humans perceive a qualitative difference between treelet and phrasal, as we see with BLEU, and (2), if the difference is perceptible, what is its magnitude relative to BLEU. If the magnitude of the difference is much larger than that of BLEU, and especially does not show convergence in the Spanish cases, then we still have a strong case for the Treelet Penalty. In fact, if human evaluators perceive a difference Spanish cases on the full data systems, the case where we show convergence, then the resulting differences could be described as the penalty value.

Unfortunately, our human evaluation data on the Treelet Penalty effect was inconclusive. Our evaluations show a strong correlation between BLEU and human evaluation, something that is attested to in the literature (*e.g.,* , the first paper on BLEU (Papineni et al., 2002), and a deeper exploration in (Coughlin, 2003)). However, the effect we were looking for – that is, a difference between human evaluations across decoders – was not evident. In fact, the human evaluations followed the differences we saw in BLEU between the two decoders very closely. Figure 11 shows data points for each data size for each decoder, plotting BLEU against human evaluation. When we fit a regression line against the data points for each decoder, we see complete overlap.[5]

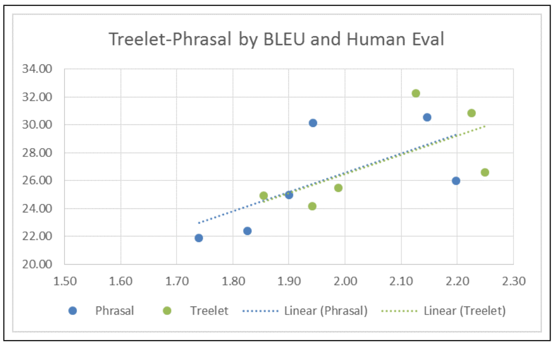


Figure 11: Scatterplot showing Treelet vs Phrasal systems across different data sizes, plotting BLEU (Y) against Human Eval scores (X)

In summary, we show a strong effect of treelet systems performing better than phrasal systems trained on the same data. That difference, however, generally diminishes as data sizes increase, and in the case of Spanish (both directions), there is a convergence in very large data sizes. These results are not completely surprising, but still are a nice systematic confirmation that linguistically informed systems really do better in lower-data environments. Without enough data, statistical systems cannot learn the generalizations that might otherwise be provided by a parse, or codified in rules. What we failed to show, at least with Spanish and German, is a confirmation of the existence of the Treelet Penalty. Given the small number of samples, a larger study which includes many more language pairs and data sizes, may once and for all confirm the Penalty. Thus far, human evaluations do not show qualitative differences between the two decoders—at least, not divergent from BLEU.

4.3 Interaction Between Decoder Type and Sentence Length

When comparing the differences between decoders, another area to pay special attention to is systematic differences in behavior as input content is varied. For example, we may expect a phrasal decoder to do better on noisier, less grammatical data than a parser-informed decoder, since in the latter case the parser may fail to parse; the failure could ripple through subsequent processes, and thus lessen the quality of the output. Likewise, a parser-informed decoder may do better on content that is short and easy to parse. If we were to do a coarse-grained separation of data into length buckets, making the very gross assumption that short equals easy-to-parse and long not, then we may see some qualitative differences between the decoders across these buckets.

To see length-based effects across decoder types, we designed a set of experiments on German and Spanish in both directions, where we separated the WMT 2010 test data into length-based word-count buckets: 0-10, 10-20, 20-30, 30-40, and 40+ words. We then calculated the BLEU scores on each of these buckets, the results for which are shown in Figures 12.

Treelet does better than phrasal in almost all conditions (except one). That is not surprising, given the results we observed in Section 4.2. What is interesting is to see how much stronger treelet

[5] Clearly, the sample is *very* small, so the regression line should be taken with a grain of salt. We would need a lot more data to be able to draw any strong conclusions.

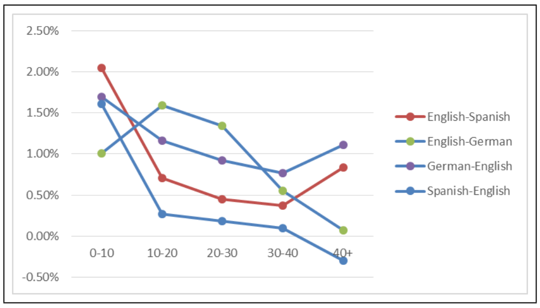


Figure 12: Treelet-Phrasal BLEU differences by bucket across language pair

performs on short content than phrasal: treelet does the best on the shortest content, with quality dropping off anywhere between 10-30 words.

One conclusion that can be drawn from these data is that treelet performs best on short content precisely because the parser can easily parse the content, and the parse is effective in informing subsequent processes. The most sustained benefit is observable in English-German, with a bump up at 10-20, and a slow tapering off thereafter. Processing the structural divergence between the two languages, especially when it comes to word order, may benefit more from a parse. In other words, the parser can help inform alignment where there are long-distance distortion effects; a phrasal system's view is too local to catch them. However, at longer sentence sizes, the absence of good parses lessen the treelet advantage. In fact, in English-German (and in Spanish-English) at 40+, there is no observable benefit of treelet over phrasal.[6]

5 The Data Gap

All Statistical Machine Translation work relies on data, and the manipulation of the data as a preprocess can often have significant effects downstream. "Data munging", as we like to call it, is every team's "secret sauce", something that can often lead to multi-point differences in BLEU. For most teams, the heuristics that are applied are fairly ad hoc, and highly dependent on the kind of data being consumed. Since data sources are often quite noisy, *e.g.,* the Web, noise reduction is a key component of many of the heuristics. Here is a list of common heuristics applied to data. Some of these are drawn from our own pre-processing, some are mentioned explicitly in other literature, in particular, (Denkowski et al., 2012).

- Remove lines containing escape characters, invalid Unicode, and other non-linguistic noise.
- Remove content that where the ratio of certain content passes some threshold, *e.g.,* alphabetic/numeric ratio, script ratio (percentage of characters in wrong form passes some threshold, triggering removal).
- Normalize space, hyphens, quotes, etc. to standard forms.
- Normalize Unicode characters to canonical forms, *e.g.,* Form C, Form KC.
- In parallel data, measure the degree of ratio of length imbalance (*e.g.,* character or word count) between source and target, as a test for misalignments. Remove sentence pairs that pass some threshold.
- Remove content where character count for any token, or token count across a sentence, exceeds some threshold (the assumption being that really long content is of little benefit due to complications it causes in downstream processing).

The point of *data cleaning* heuristics is to increase the value of training data. Each data point that is noisy increases the chance of learning something that could be distracting or harmful. Likewise, each data point that is cleaned reduces the level of data sparsity (*e.g.,* through normalizations or substitutions) and improves the chances that the models will be more robust. Although it has been shown that increasing the amount of training data for SMT improves results (Brants et al., 2007), not all data is beneficial, and clean data is best of all.

Crucially, most data munging is done through heuristics, or rules, although thresholds or constraints can be tuned by data. A more sophisticated example of data cleaning is described in (Denkowski et al., 2012) where the authors used machine learning methods for measuring quality estimation to select the "best" portions of a corpus. So, rather than training their SMT on an entire corpus, they trained an estimator that selected

[6] The bump up at 40+ on English-Spanish and German-English is inexplicable, but may be attributable to the difficulty that either decoder has in processing such long content. There is also likely an interaction with statistical noise cause by such small sample sizes.

the best portions, and used only those. In their entry in the 2012 WMT competition, they used only 60% of the English-French Gigaword corpus[7] and came in first in the shared translation task for the pair.

Another important aspect of data as it relates to SMT is task-dependence: what domain or genre of data will an SMT engine be applied to? For instance, will an SMT engine be used to translate IT content, news content, subtitles, or Europarl proceedings? If the engine itself is trained on data that is dissimilar to the desired goal, then results may be less than satisfying. This is a common problem in the field, and a cottage industry has been built around customization and domain-adaptation, *e.g.,* (Moore and Lewis, 2010; Axelrod et al., 2011; Wang et al., 2012). In general, the solution is to adapt an SMT engine to the desired domain using a set of seed data in that domain.

A more difficult problem is when there is very little parallel data in the desired domain, which is a problem we will look at in the next section.

5.1 Preprocessing Data to Make it Match

A little over a year ago, Facebook activated a translation feature in their service, which directly called Bing Translator. This feature has allowed users to translate pages or posts not in their native language with a *See Translation* option. An example is shown in Figure 13.

The real problem with translating "FB-speak", or content from virtually any kind of social media, is the paucity of parallel data in the domain. This flies in the face of the usual way problems are tackled in SMT, that is, locate (lots of) relevant parallel data, and then train up a decoder. Outside of a few slang dictionaries, there is almost no FB-like parallel content available.

Given the relatively formal nature of the text that most of our engines are trained on, the mismatch between FB content and our translation engines often led to very poor translations. Yet, given the absence of in-domain parallel data, it was not possible for us to train-up FB-specific SMT engines. We realized that our only option was to somehow manipulate the input to make it look more like the content we trained our engines on. Effectively, if we treated "FB-speak" as a dialect of the source language, we could use distributional queues of dialect-specific content to find the counterparts in the majority dialect.

Regex	Output
frnd[sz]	friends
plz+	please
yess*	yes
be?c[uo][sz]	because
nuff	enough
wo?u?lda	would have
srr+y	sorry

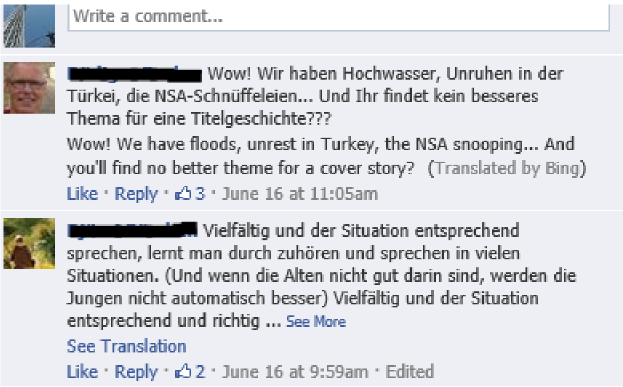
Table 5: Some example regexes to "fix" FaceBook content

Table 4 gives some examples of FB content on the left, and the more formal representation of the same on the right. The reader will note some systematic characteristics of the FB content as compared to the formal content (see also (Hassan and Menezes, 2013)). Given the absence of parallel training data, we could "correct" the FB content to make it look more like English, and then translate the "corrected" English through our engines.

Our first inclination was to examine the logs of the most frequent words being translated by FB users and use string substitutions or regexes (regular expressions) to effect repairs. We arrived very quickly at a large set of simple repairs like those shown in Table 5. We were able to achieve greater than 97% precision using a large table of substitutions for the most common translations (against a held-out set of FB content). However, there were two problems with the approach: (1) recall was relatively low, at 52.03%, and (2) the solution was not easily scalable to additional languages and scenarios.

To address these two deficiencies, we sought a more data-driven approach. But we had to be creative since our standard "hammer" of parallel data did not exist. Our intuition was that there were distributional regularities in the FB content that could help discover a mapping for a given target word, *e.g.,* the distribution of *plzzz* in the FB content would allow us to discover that it distributes similarly to *please* in our non-FB content. Hany Hassan developed a TextCorrector tool that is, as he put it (Hassan and Menezes, 2013), "based on constructing a lattice from possible normalization candidates and finding the best normalization sequence according to an n-gram language model using a Viterbi decoder", where he developed an

[7] The English-French Gigaword corpus is described in (Callison-Burch et al., 2009)



Write a comment...

[redacted] Wow! Wir haben Hochwasser, Unruhen in der Türkei, die NSA-Schnüffeleien... Und Ihr findet kein besseres Thema für eine Titelgeschichte???

Wow! We have floods, unrest in Turkey, the NSA snooping... And you'll find no better theme for a cover story? (Translated by Bing)

Like · Reply · 👍3 · June 16 at 11:05am

[redacted] Vielfältig und der Situation entsprechend sprechen, lernt man durch zuhören und sprechen in vielen Situationen. (Und wenn die Alten nicht gut darin sind, werden die Jungen nicht automatisch besser) Vielfältig und der Situation entsprechend und richtig ... See More

See Translation

Like · Reply · 👍2 · June 16 at 9:59am · Edited

Figure 13: Two Facebook posts: the first translated, the second showing the *See Translation* option

FB Speak	English Translation	Comment
goooood morniiing	good morning	Extended characters for emphasis or dramatic effect
wuz up bro	What's up brother	"Phonetic" spelling to reflect local dialect or usage
cm to c my luv	Come to see my love	Remove vowels in common words, sound-alike sequences
4get, 2morrow	forget, tomorrow	Sound-alike number substitution
r u 4 real?	Are you for real?	Sound-alike letter and number substitutions
LMS IDK ROFL	Like my status I don't know Rolling on the floor laughing	Single 'word' abbreviations for multi-word expressions

Table 4: FB Speak with English references

"unsupervised approach to learn the normalization candidates from unlabeled text data." He then used a Random Walk strategy to walk a contextual similarity graph. The two principal benefits of this approach is that it did not require parallel training data—two large monolingual corpora are required, one for the "noisy" data (*i.e.,* FB content) and one for the clean data (*i.e.,* our large supply of language model training data)—nor did it require labeled data (*i.e.,* , the algorithm is unsupervised). After several iterations over very large corpora (tens of millions of sentences) he arrived at a solution that had comparable precision to the regex method but had much higher recall. The best iteration achieved 96.51% precision (the regex approach achieve 97.07% precision) and 72.38% recall (regex: 52.03%).[8] Crucially, as the size of the data increases, the TextCorrector continues to show improvement.

The end result was a much better User Experience for FB users. Rather than badly mangled translations, or worse, no translations at all, users get translations generated by our standard, very large statistical engines (for English source, notably, our *treelet* engines). An example English source string is shown in Table 6, with translations shown for both the corrected and uncorrected source.

6 Conclusions and Future Directions

A crucial lesson from the work on the FB corrections described in Section 5.1 is its analog to Machine Learning as a whole: rule-based approaches often achieve very high precision, but often at the sacrifice of recall. The same is true in Machine Translation: rule-based MT is often more accurate when it was accurate, resulting in more precise and grammatical translations. However, it tends to be somewhat brittle and does not do as well on cases not explicitly coded for. SMT, on the other hand, tends to be more malleable and adaptable, but often less precise. Tapping rule-based approaches in a statistical framework can really give us the best of both worlds, giving us higher precision *and* higher recall.

Finding an appropriate mix is difficult, though. As in the case of parsing, we can see how errors can substantially degrade translation quality, especially if we only consider the single best analysis. By making our analysis components as robust as possible, quantifying our degree of certainty with scoring mechanisms, and preserving ambiguity of the analysis, we can achieve a better return on in-

[8] For a complete description of TextCorrector, please see (Hassan and Menezes, 2013).

Language	Unrepaired	Repaired
Original English	i'l do cuz ma parnts r ma lyf	I'll do because my parents are my life
To Italian	i ' l fare cuz ma parnts r ma lyf	lo far perch i miei genitori sono la mia vita
To German	i ' l tun Cuz Ma Parnts R Ma lyf	Ich werde tun, weil meine Eltern mein Leben sind
To Spanish	traer hacer cuz ma parnts r ma lyf	voy a hacer porque mis padres son mi vida

Table 6: One English FB sentence with and without normalizations, translated to various languages

vestment. Making this linguistic information be included *softly* as features is a powerful way of surfacing linguistic generalizations to the system while not forcing its hand.

Some of the greatest successes in mixing linguistic and statistical methods have been in syntax. There is much ground to cover still. Morphology is integrated weakly into current SMT systems, mostly as broad features (Jeong et al., 2010) though sometimes with more sophistication (Chahuneau et al., 2013). Better integration of morphological features could have great effect, especially in agglutinative languages such as Finnish and Turkish.

Deeper models of semantics present a rich challenge to the field. As we proceed into deeper models, picking the correct representation is a significant issue. Humans can generally agree on words, mostly on morphology, and somewhat on syntax. But semantics touches on issues of meaning representation: how should we best represent semantic information? Should we attempt to faithfully represent all the information in the source language, or gather only a simple model that suffices to disambiguate information? Others are focusing on lexical semantics using continuous space representations (Mikolov et al., 2013), a softer means of representing meaning.

Regardless of the details, one point is very clear: future work in MT will require dealing with data. Systems, whether statistical or rule-based, will need to work with and learn from the increasing volumes of information available to computers. Effective hybrid systems will be no exception – tempering the keen insights of experts with the noisy wisdom of big data from the crowd holds great promise.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of EMNLP*, pages 355–362.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic, June. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece, March. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden, July. Association for Computational Linguistics.

Marine Carpuat and Michel Simard. 2012. The trouble with smt consistency. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 442–449, Montréal, Canada, June. Association for Computational Linguistics.

Victor Chahuneau, Noah A. Smith, and Chris Dyer. 2013. Knowledge-rich morphological priors for bayesian language models. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1206–1215, Atlanta, Georgia, June. Association for Computational Linguistics.

Deborah A. Coughlin. 2003. Correlating automated and human assessments of machine translation quality. In *Proceedings of MT Summit IX*, New Orleans, Louisiana, USA, September. The Association for Machine Translation in the Americas (AMTA).

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The CMU-Avenue French-English Translation System. In *Proceedings of the NAACL 2012 Workshop on Statistical Machine Translation*.

Mireia Farrús, Marta R. Costa-jussá, and Maja Popovic. 2012. Study and correlation analysis of linguistic, perceptual and automatic machine translation evaluations. *Journal of the American Society for Information Science and Technology*, 63(1):174–184, January.

Hany Hassan and Arul Menezes. 2013. Social text normalization using contextual graph random walks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-hmm-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of EMNLP*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Minwoo Jeong, Kristina Toutanova, Hisami Suzuki, and Chris Quirk. 2010. A discriminative lexicon model for complex morphology. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA-2010)*.

Kevin Knight. 2013. Tutorial on decipherment. In *ACL 2013*, Sofia, Bulgaria, August.

Arul Menezes and Stephen D. Richardson. 2001. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Association for Computational Linguistics*.

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.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

Robert C. Moore and William D. Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden, July.

Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguisitics*, 30(4):417–449, September.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st ACL*, 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 ACL*, Philadelphia, PA.

Chris Quirk and Simon Corston-Oliver. 2006. The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 62–69, Sydney, Australia, July. Association for Computational Linguistics.

Chris Quirk and Arul Menezes. 2006. Dependency Treelet Translation: The convergence of statistical and example-based Machine Translation? *Machine Translation*, 20:43–65.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Wei Wang, Klaus Macherey, Wolfgang Macherey, Franz Och, and Peng Xu. 2012. Improved domain adaptation for statistical machine translation. In *Proceedings of AMTA*.

Warren Weaver. 1955. Translation. In William N. Locke and A. Donald Booth, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Massachussets.

Hao Zhang, Licheng Fang, Peng Xu, and Xiaoyun Wu. 2011. Binarized forest to string translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 835–845, Portland, Oregon, USA, June. Association for Computational Linguistics.