Mach Translat (2011) 25:317–339
DOI 10.1007/s10590-011-9108-7

Syntactic discriminative language model rerankers for statistical machine translation

Simon Carter · Christof Monz

Received: 28 December 2010 / Accepted: 12 August 2011 / Published online: 1 September 2011

Abstract This article describes a method that successfully exploits syntactic features for n-best translation candidate reranking using perceptrons. We motivate the utility of syntax by demonstrating the superior performance of parsers over n-gram language models in differentiating between Statistical Machine Translation output and human translations. Our approach uses discriminative language modelling to rerank the n-best translations generated by a statistical machine translation system. The performance is evaluated for Arabic-to-English translation using NIST's MT-Eval benchmarks. While deep features extracted from parse trees do not consistently help, we show how features extracted from a shallow Part-of-Speech annotation layer outperform a competitive baseline and a state-of-the-art comparative reranking approach, leading to significant BLEU improvements on three different test sets.

Keywords Statistical machine translation · Discriminative language models · Syntax

1 Introduction

Language models (LMs), alongside translation models, form the core of modern Statistical Machine Translation (SMT) systems, whether they be phrase-based

This work is a revised and substantially expanded version of (Carter and Monz 2009) and (Carter and Monz 2010).

S. Carter (✉) · C. Monz
ISLA, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
e-mail: s.c.carter@uva.nl

C. Monz
e-mail: c.monz@uva.nl

(Koehn et al. 2003) or hierarchical systems (Chiang 2005, 2007). A language model's primary job in an SMT system is to check and enforce the fluency of a translation candidate without any knowledge of the source sentence being translated. In so doing, the language model impacts on word order decisions and translation selection decisions.

The most commonly used LMs are word n-gram models, which make the Markovian assumption and base the probability of a word following a prefix string on a limited context of the previous $n - 1$ words. Given that n-gram LMs utilise a limited lexical context when deciding on the overall fluency of a sentence, the use of syntax, which contains deeper, and consequently long-range dependency information, is intuitively appealing.

We motivate the use of syntax by first demonstrating that parsers are better able to discriminate between fluent and disfluent English than a large four-gram language model. We then show that the probability output by a parser cannot differentiate between varying degrees of fluent English, because if often tries to assign sensible structure to disfluent word sequences. As a result of the initial experiments presented in this article, we propose a syntactic discriminative language model, which allows for the use of finer-grained syntactic features, to be used in a n-best reranking setting. Together with this model, we suggest the extraction of features from a novel non-context-aware Part-of-Speech (POS) tagger. Accordingly, we aim to integrate syntactic features that represent the actual word sequence, leading to models that better discriminate between disfluent, machine-generated language and human-produced sentences.

The remainder of this article is structured as follows: we discuss related work in Sect. 2. We then examine the ability of a parser to discriminate between sentences of varying degrees of fluency in Sect. 3. We introduce discriminative LMs in Sect. 4, present the perceptron model used in Sect. 4.1, and describe the syntactic features in Sect. 4.2. Experimental results are reported in Sect. 5. We discuss the findings in detail in Sect. 6, and conclude in Sect 7.

2 Related work

One manner to overcome the data sparseness problem of n-gram LMs has been the generalisation of existing data to encapsulate greater information that go beyond the surface form. Class-based models (Brown et al. 1992) generalise from the word form to an abstract representation based on word clusters. These models are often used during decoding via interpolation with a standard lexical model. The class-based models are further generalised via Factored language models (FLMs) (Bilmes and Kirchhoff 2003; Birch et al. 2007; Koehn and Hoang 2007). Here, each word is represented by as many different forms as are required, from morphological information to supertags and partial parses (Birch et al. 2007). Both of these move beyond the surface word form, but stay within the n-gram framework.

A study of the use of a range of generative syntactic models was undertaken by Och et al. (2003, 2004), who performed n-best reranking as we have done. Syntactic features were not successful. The approach of Post and Gildea (2008) integrates a parser as language model into the decoding framework, but they are unable to outperform the

baseline. Unlike the two previous approaches which attempted to utilise full parse trees for improving SMT output, the utility of shallow parse information has been demonstrated by Hasan et al. (2006) for the translation of speech corpora. Supertagging, lightweight dependency analysis, a link grammar parser and a chunk parser are used to rerank n-best lists within a log-linear framework.

Shen et al. (2004) were the first to use a perceptron-like algorithm in a small-scale application of reranking SMT n-best lists. They used the algorithm to optimise weights for a small number of features (tens instead of millions). The use of perceptron-type algorithms with millions of features for SMT has been explored by Arun and Koehn (2007). They examine the use of online algorithms for the discriminative training of a phrase-based SMT system. In this article we focus on the use of perceptrons for reranking using only target-side syntactic information. Other researchers have attempted to examine the training of specific features types; from the training of reordering features (Tillmann and Zhang 2006; Chang and Toutanova 2007), translation model features (Blunsom et al. 2008), independent target- and source-side features (Chiang et al. 2009), and both translation and language model features combined (Shen et al. 2004; Liang et al. 2006; Watanabe et al. 2007; Chiang et al. 2008).

The work most closely related to ours is the discriminative syntactic LM proposed in (Collins et al. 2005). The work presented in this article differs in two important aspects. First, we focus on the use of syntactic features in SMT. Second, we propose the use of a simple POS tagger, which gives significant improvements over a 1-best baseline and competing state-of-the-art reranker. Li and Khudanpur (2008) apply the framework of Roark et al. (2007) to create the first large-scale discriminative language model to SMT for reranking. Using a standard n-gram feature set, they outperformed the 1-best output of their baseline SMT system. They focus on the application of n-gram-only models to SMT and the use of data filtering thresholds.

This article extends our previous work in Carter and Monz (2009, 2010) in three ways; first by expanding upon the original experiments in Carter and Monz (2009) with two new additional test sets, and three higher order LMs, demonstrating the validity of the results and conclusions drawn. Second, we motivate the work first presented in Carter and Monz (2010) by giving concrete examples of where parsers go wrong, motivating the best-performing S-POS reranker. Finally, we present four new deep, syntactic reranking, models, and provide a more detailed analysis of both deep and shallow syntactic features for reranking SMT output.

3 Parsing ungrammatical English

In this section, we examine the ability of a parser to differentiate between fluent and disfluent English. Specifically, we look at two tasks. The first, in Sect. 3.3, is differentiating between SMT output and human-produced English. If a state-of-the-art parser is unable to do this task as well as a standard LM, then it is not suitable for use within an SMT system, as suggested by Och et al. (2004). The second, harder task, reported in Sect. 3.4, is to differentiate between varying degrees of fluent sentences produced by an SMT system. This second set of experiments is representative of a standard reranking task.

3.1 Parser

We used an implementation of Collins Model 2 (CM2) by Bikel (2002) to provide parses and parse probabilities. CM2 uses four different probability distributions for assigning probabilities to (i) head labels, (ii) sibling labels and head tags, (iii) sibling head words, and (iv) subcategorization frames. We chose Model 2 above Model 1 because of the higher reported Labeled Recall and Precision scores (Collins 1997). The parser was trained on sections 02-21 of the Wall Street Journal portion of the Penn Tree Bank (PTB) (Marcus et al. 1994), about 40,000 sentences, and tested on section 23, about 2,500 sentences. Because we are applying the parser to lowercased SMT output, we lowercase the parser training data.

We compare the parser against a large four-gram language model. The SRILM toolkit (Stolcke 2002) with modified Kneser-Ney smoothing was used to build the target four-gram language model using the AFP and Xinhua portions of the English Gigaword corpus (LDC2003T05) and the English side of the bitext.

3.2 Experimental set-up

In this section, we provide details of the experimental set-up for our experiments analysing the ability of parsers to differentiate between fluent and disfluent English.

Moses was used as a state-of-the-art baseline SMT system for reporting experimental results (Koehn et al. 2007). It is a phrase-based MT system using stacks to organise partial translation candidates. The parameters used for the experiments discussed here are stack size of 100, distortion limit of 6, and phrase table limit of 20. We utilise lexicalised reordering along with the standard Moses phrase tables scores and a language model.

To build the phrase table and language model, we used five corpora distributed by the Linguistic Data Consortium (LDC), totaling 300K sentence pairs. The parallel text includes Arabic news LDC2004T18, automatically extracted parallel text LDC2007T08, eTIRR news LDC2004E72 and translated Arabic treebank data LDC2005E46. Alignments were extracted using the GIZA++ toolkit (Och and Ney 2000). Minimum Error Rate Training (MERT) (Och 2003) was used for optimising the parameters of the Moses baseline SMT system.

For Arabic-to-English translation, performance is evaluated using NIST's MT-Eval benchmarking sets from 2002 through to 2006 (henceforth referred to as MT02, MT03, MT04, MT05 and MT06). Statistics for each set (#source sentences/#refs): MT02 (1043/4), MT03 (663/4), MT04 (1353/5), MT05 (1056/5), MT06(1796/4). Sets MT02 and MT03 were used for development.

3.3 Discriminating between SMT and human translations

In this section we present results on the discrimination of the SMT output and reference translation using a parser or LM probabilities. Formally, our experiment is composed of a set of SMT output sentences and their respective reference translations, which in our case is 4 each. In this work, a specific SMT output sentence with its respective

Table 1 Percentage of references with higher probability than respective SMT output for the test sets MT04, MT05 and MT06

Model	No normalisation			Normalised		
	MT04	MT05	MT06	MT04	MT05	MT06
LM4	13.58	28.05	15.24	23.08	30.78	30.68
LM5	15.63	34.44	17.2	26.92	35.72	34.28
LM6	16.09	33.62	17.35	27.27	36.55	34.58
CM2	**34.09**	**49.1**	**33.6**	**61.09**	**54.85**	**63.37**

We present normalised and unnormalised results
We highlight in bold the best scores

references are referred to as a 'sentence set'. Unlike in our previous work (Carter and Monz 2009), no thresholding based on SMT output sentence length is applied. We define the accuracy of a model to be the percentage of references which are assigned an equal or higher parser/LM probability than the MT output.

We show in Table 1 the percentage of reference sentences which were assigned a higher or equal probability to the respective SMT output by each of the different models on the three different test sets. All results are reported using simple length normalisation, where the log probability is divided by the sentence length, as we found this to have a positive impact. The CM2 parser outperforms the n-gram LMs. When using length normalisation, the parser performs better than chance, while the LMs continue to perform worse than 50%. While there is a slight increase in both unnormalised and normalised language model scores by using higher order LMs, the parser still achieves accuracy results between 1.5 (MT05) and 2.24 (MT04) times better than the corresponding best LM6 scores.

To see to what extent these findings hold for SMT sentences of different degrees of fluency, where fluency is approximated by BLEU, the standard evaluation metric for SMT systems (Papineni et al. 2002), we bucketed the reference sentences by the sentence-level BLEU score of the corresponding SMT output.[1] We would expect that SMT output which has a higher BLEU score is harder to differentiate from the human-produced references (cf. Kulesza and Shieber 2004). Results are displayed in Fig. 1b, c and d. The correlation between BLEU range and model accuracy is measured by applying linear regression. Unsurprisingly, as shown by the linear regression lines, model accuracy appears to decrease as BLEU increases for all three test sets; the better the SMT output, the harder it is for both models to differentiate between good and bad English. Note, however, that the relative classification accuracy remains the same between the models.

To take a deeper look into which sentences the models were having problems with, we bucketed the SMT translations by their length, and examined the classification accu-

[1] Though there are reported issues with the use of BLEU (cf. Callison-Burch et al. 2006), a manual evaluation of the SMT output confirms a correlation between the sentence-level BLEU scores and fluency of the translations. Furthermore, as this is the established metric on which SMT systems are optimised towards and evaluated against, we believe it to be a sufficient proxy for fluency.

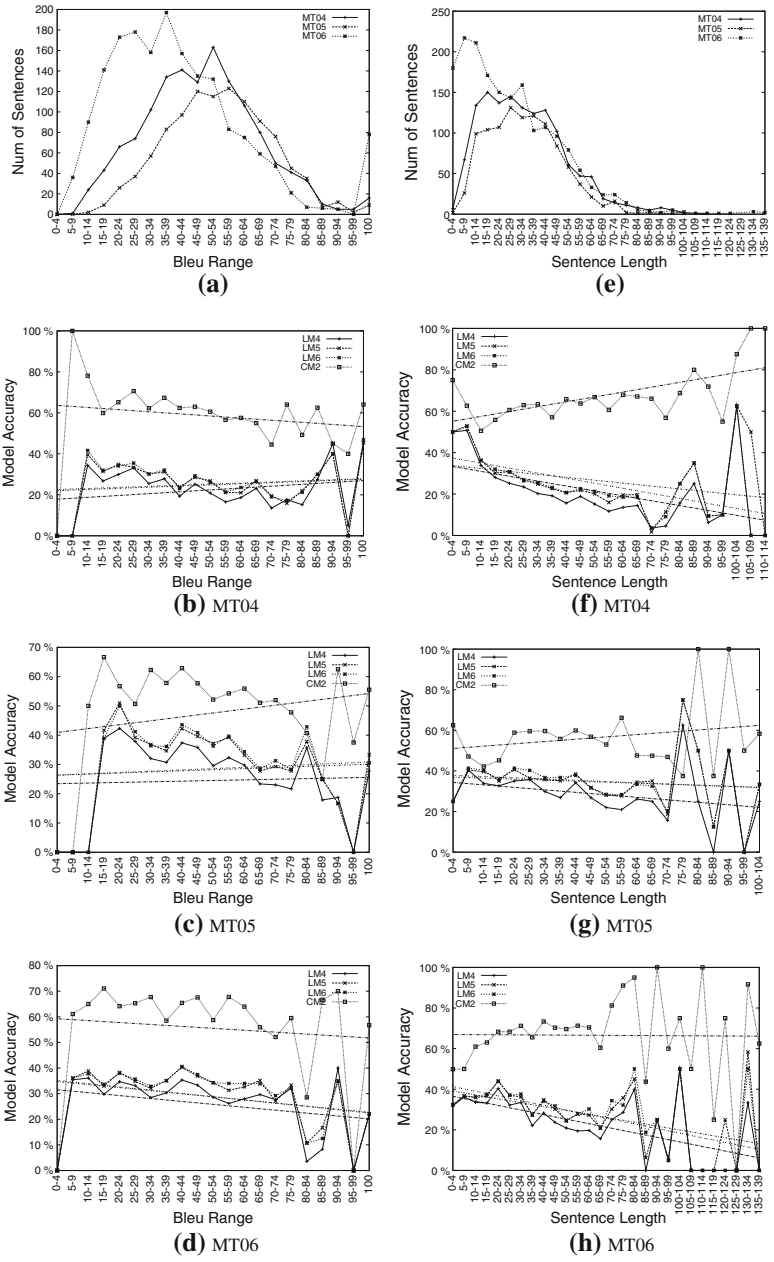


Fig. 1 Breakdown of model performance in discriminating between SMT output and human-produced references using length-normalised probabilities of a 4-, 5- and 6-gram LM and state-of-the-art parser. **a** shows the number of sentences per BLEU bin for each of the three test sets. **e** shows the number of SMT sentences in each length bin. **b**, **c** and **d** give a breakdown of classification accuracy by BLEU score of SMT sentence for MT04, MT05 and MT06 test sets. **f**, **g** and **h** show the classification accuracy of the length-normalised models by length of the SMT sentences for the three test sets. **b**, **c**, **d** and **f**, **g**, **h** have linear regression lines drawn for both models showing the underlying trends. All BLEU scores in these figures are sentence-level BLEU

Table 2 Average Pearson correlation coefficient between the two different models and BLEU rankings for each of the MT04, MT05 and MT06 test sets

Model	MT04	MT05	MT06
LM4	−0.045	0.066	−0.027
CM2	−0.027	0.036	−0.046

racy for each bucket for each model. The results are displayed in Fig. 1f, g and h. Even for short reference sentences in the range of 1–4 words, the parser still outperforms the n-gram LMs. Moreover, as the regression lines show, the parser performs better as sentence length increases, in contrast to the LMs, whose performance decreases with longer translations. The parser is able to exploit deep syntactic structures that capture long-range information within the sentence to assign it a parse whose probability better reflects the fluency of the translation, whereas the n-gram LMs, limited to a fixed history, are unable to utilise this information.

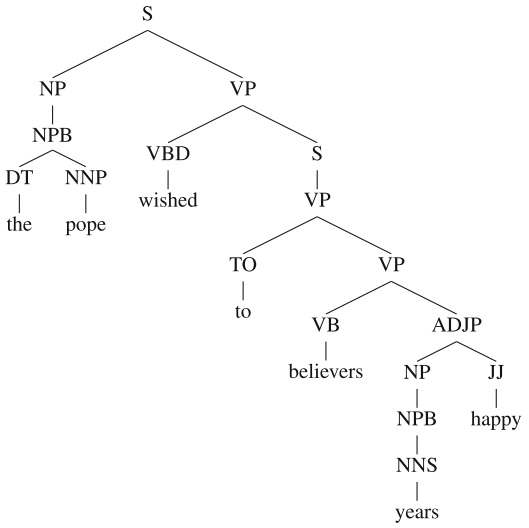
These results are interesting, as parsers perform worse on standard Labeled Recall and Labeled Precision measures as sentence length increases (McDonald 2007). This demonstrates that these measures—the evaluation metrics which parsers are traditionally developed and optimized towards—are not necessarily indicative of a parser's ability to differentiate between SMT output and human-produced translations.

3.4 Correlating parser scores with translation quality

Considering a parser's ability to better discriminate between SMT output and human translations, we propose to use parsers for reranking. As a first step we examine the correlation between the probabilities assigned to a parser in an n-best list and smoothed BLEU (Lin and Och 2004). We use the same SMT system and models as before. For each source sentence we output an n-best list of translation candidates. We score each sentence in the n-best list according to smoothed BLEU, and score each sentence with both the n-gram language model and parser. We convert these scores to ranks, tied where necessary, and compute the Pearson correlation coefficient between the BLEU and model rankings. The correlation coefficients are then averaged over the entire test set. Results are reported in Table 2. There is no correlation between the rankings assigned to the n-best lists by either n-gram or parser models and BLEU.

3.5 Analysis

While the parser is able to discriminate between SMT output and human-produced sentences, the results reported in Sect. 3.4 highlight the difficulty in using the probabilities for discriminating between sentences of varying degrees of fluency and grammaticality. A look to the literature offers some potential explanations. In analysing why the inclusion of scores from a parser—similar to the used in this article—during decoding did not help improve BLEU scores (Post and Gildea 2008), the authors argue that the use of a single probability to represent the quality of a single parse is too coarse a met-



Type	Sentence	LM4	CM2
SMT	the pope wished to believers years happy	**-26.61**	-5.5
REF 1	pope wishes worshipers happy new year	-22.75	**-7.13**
REF 2	pope wishes believers happy year .	-24.21	-6.01
REF 3	pope wishes the faithful a happy year	-25.86	-5.72
REF 4	pope wishes happy year for the faithful	-23.3	-5.92

Fig. 2 We show the parse over an SMT translation of the MT04 source segment 139. In the table we show the same translation, along with the four human-produced reference translations. In the two final columns we print the length-normalised log probability assigned to each sentence by the four-gram LM (LM4) and Collins Model 2 (CM2) parser. The parser assigns the SMT output a higher normalised log probability than the four reference translations, while the n-gram LM assigns the sentence the lowest normalised log probability. We highlight in bold the lowest scores output by both models

ric for the successful exploitation of parse tree information. These results corroborate the earlier experiments of Och et al. (2004), which demonstrate the lack of utility of a generative parser for reranking. The results from our averaged correlation experiments in Sect. 3.4 support the literature in showing that the direct use of a single probability to rank an n-best list does not lead to improvements in BLEU.

To give an example of why this is the case, we show in Fig. 2 a parse by Collins Model 2 over a 1-best sentence output from the MT04 test set. Here, the length-normalised log probability assigned by the parser to this disfluent sentence was higher than those assigned to all four reference sentences. The parser incorrectly tags 'believers' as a verb instead of a noun. The parser does this to obtain a good structure; the cost of correctly tagging 'believers' as a noun is too prohibitive. A quick glance at the WSJ training corpus hints at a reason why: out of 21K instances of TO, its nearest rightmost sibling is a VP 13,001 times, and an NP 8,460 times, making the former more likely. In assuming the input is a well-formed fluent English sentence, assigning the most likely parse leads to good high-level structures that mask disfluency at the local level. In comparison, the n-gram language model, which makes only local decisions, correctly assigns the sentence a lower length-normalised probability than its references.

Looking at the example parse, one can formulate a number of different features that could be used to distinguish between a fluent and disfluent sentence. A lexicalised parent-head-based feature, similar to that used in Collins et al. (2005), could determine that a VP headed by 'believers' is unlikely. Furthermore, a shallow POS n-gram sequence extracted from the parse could show that the trigram VB NNS JJ is improbable.

There are many features, deep or shallow, that could be extracted and learnt from the parse that lead to potential gains in a reranking setting. Our solution is to examine the potential benefits of multiple syntactic features exploited in a discriminative language model framework. Such a framework would allow us to conduct a thorough investigation of the different types of syntactic information extractable from a full parse tree in a computationally practical manner.

4 Syntactic discriminative language models (DLMs)

DLMs consist of a function $\phi(\cdot)$ that maps a sentence onto a feature space and weight vector w (Roark et al. 2007). For training, negative and positive examples are supplied to the DLM for learning the weight vector. The weight vector and feature function is then used to assign a non-probabilistic score to an unseen sentence.

A benefit of using discriminative techniques over generative models is that standard generative LMs are trained exclusively on well-formed English. Given the large feature space they operate in, the accurate assignment of probabilities to unseen events is a difficult problem, and has been a major area of research for the past sixty years (for a detailed overview on language modelling and smoothing techniques, see (Chen and Goodman 1998). Discriminative models are trained with positive and negative examples, and therefore learn to assign negative weights to harmful features, without having to infer this from positive data only.

Different parameter estimation methods to estimate the weight vector w for a DLM have been previously examined in the Automatic Speech Recognition (ASR) domain (Roark et al. 2004b, 2007). These include optimising the log-likelihood under a log-linear model, a batch algorithm which requires processing all the data before outputting a weight vector as an answer, and approximating a 0/1 loss through the perceptron update rule, and an online algorithm which examines and updates the parameter vector sequentially. The reader is referred to (Roark et al. 2004b; Emami et al. 2007) for a discussion on the benefits of the log-linear model and the perceptron. Given that this article examines the use of a syntactic feature space, which is larger than an already large n-gram feature space, and that perceptrons perform feature selection as a consequence of its learning procedure, we opt to use the perceptron algorithm.

4.1 Perceptron

The perceptron, proposed by Rosenblatt (1958), is an online error minimisation learner that, assuming linearly separable data, can theoretically converge to a solution that perfectly classifies the data (Freund and Schapire 1999). The perceptron has been successfully applied to parse reranking (Collins and Duffy 2002), document reranking

Fig. 3 The standard perceptron algorithm

Perceptron

1: $w \leftarrow 0$
2: **for** $t = 1$ to T **do**
3: **for** $i = 1$ to N **do**
4: $y^i \leftarrow ORACLE(x^i)$
5: $z^i \leftarrow argmax_{z \in x^i} \phi(z) \cdot w$
6: **if** $z^i \neq y^i$ **then**
7: $w \leftarrow w + \phi(y^i) - \phi(z^i)$
8: **end if**
9: **end for**
10: **end for**
11: **return** w

for IR (Crammer and Singer 2001; Elsas et al. 2008; Chen et al. 2009), ASR reranking (Roark et al. 2004b; Collins et al. 2005; Singh-Miller and Collins 2007), and finally to SMT translation reranking (Shen et al. 2004; Li and Khudanpur 2008), where Chinese–English translation systems were significantly improved.

4.1.1 Algorithm

The standard perceptron algorithm is shown in Fig. 3. The algorithm takes as input a set of n-best lists X, and an oracle function $ORACLE(x^i)$ that determines the best translation (oracle best) for each of the n-best lists x^i according to the BLEU metric. As DLMs make comparisons at the sentence level, we use sentence-level BLEU with additive smoothing (Lin and Och 2004). While there are discrepancies between sentence- and corpus-level BLEU, we find sentence-level BLEU sufficient for reranking SMT. T defines the number of iterations and N defines the size of the test set, which in our case is the number of n-best lists. The algorithm iterates over the n-best lists in a sequential manner (lines 2 and 3). If the selected hypothesis and oracle best sentence match, the algorithm continues to the next n-best list. Otherwise, the weight vector is updated (line 7). Finally, it returns a weight vector as its solution (line 11).

To use the weight vector returned by the perceptron algorithm, each sentence z in an n-best list is scored as in (1):

$$S(z) = \beta\phi_0(z) + w \cdot \phi(z) \tag{1}$$

The SMT model score for each translation hypothesis $\phi_0(z)$ is weighted by β. Roark et al. (2007) argue that while it is possible to include $\phi_0(z)$ as a feature of the perceptron model, this may lead to under-training, so we adhere to the convention of using a fixed value for β.

To score an n-best list x^i, we use the weight vector returned by the perceptron to assign a score to each sentence and select the best one, as in (2):

$$z^* = \text{argmax}_{z \in \text{GEN}(x^i)} S(z) \tag{2}$$

4.1.2 Variants

A shortcoming of the perceptron is that it can be unstable if the training data is not linearly separable. A number of solutions have been proposed in the literature. One solution is to use an averaged perceptron (Freund and Schapire 1999), where the parameter vector w output by the algorithm is averaged over each instance $w_{avg} = \Sigma_{t=1}^{T} \Sigma_{i=1}^{N} \frac{w_t^i}{N \cdot T}$. Another solution is the pocket perceptron (Gallant 1999; Collins and Duffy 2002), where the weight vector returned is the one that correctly classifies the most training instances in a row, keeping an optimal model in its 'pocket'. A third solution, called the committee or voting perceptron, keeps a cache of optimal models, sorted by their success counts (Roark et al. 2004a; Elsas et al. 2008). The cache sizes differentiate the voting and committee perceptron, with the voting perceptron using the best cached model, and the committee perceptron utilising the top-n cached models. As previous published work on using perceptrons for reranking SMT output utilised the average perceptron (Li and Khudanpur 2008), we also use this model.

4.2 Features

In examining different syntactic features, we distinguish between deep features—those extractable from a syntactic annotation layer that goes beyond pre-terminals—and shallow features which require only POS tags. In Sect. 4.2.1, we outline the different toolkits that are used to extract features. In Sect. 4.2.2, we detail the features used that can only be extracted from a full parse tree. In Sect. 4.2.3 we explain the features that can be extracted from a POS tagger. In Table 3, we list the features we examine in this article, and provide references for those feature types that have been explored, albeit for different tasks, in either an SMT or ASR setting.

4.2.1 Annotation layers

In this section, we outline the different toolkits and resulting output annotation layers from which we extract our syntactic features. Some of the features examined can only be extracted from a full parse tree, while others can be extracted from either parsers or taggers.

It is interesting to see whether it is beneficial to use features extracted from full parse trees as opposed to those extracted from a POS tagger or other shallow representation. Using parsers allows us to extract global features that relay deep structural information. Unfortunately, they are slow and memory-intensive, and may fail to return a parse for long sentences, as they have $O(n^3)$ complexity in relation to sentence length. On the other hand, POS taggers, while outputting no deep syntactic information, are more efficient and robust, as they always output a complete POS sequence

CRF tagger (*CRF*) We used Xuan-Hieu Phan's implementation of a Conditional Random Field tagger as our state-of-the-art POS tagger.[2]

[2] Available at: http://crftagger.sourceforge.net.

Table 3 Syntactic feature types examined in this article

Feature type	Field	Task	Citation	Helpful
SEQ-B & SEQ-C	ASR	Reranking	Collins et al. (2005)	Yes
CFG	ASR	Reranking	Collins et al. (2005)	Yes
HEAD	ASR	Reranking	Collins et al. (2005)	Yes
T-DEPTH	SMT	Difficult-to-translate phrase localisation	Mohit and Hwa (2007)	Yes
UNIT-P	–	–	–	–
NT-C	SMT	Reranking/decoder features	Och et al. (2003); Chiang et al. (2009)	No/yes
NO-C	SMT	–	–	–
POS	ASR / SMT	Reranking/FLM features	Collins et al. (2005); Birch et al. (2007)	Yes
VERBAGR	–	–	–	–
POSNUM	–	–	–	–
NOPOS	SMT	Reranking/decoder features	Och et al. (2003); Chiang et al. (2009)	No

During experimentation different feature combinations are examined. Where a feature has been previously used, we list the field, task and citation, and whether or not it proved useful

Simple tagger (*S-POS*) We also use a simple maximum likelihood POS tagger, which assigns to each word the most likely tag according to a training set, regardless of any context. The simple model does not use any smoothing, meaning that out-of-vocabulary items are simply assigned ⟨UNK⟩ as their tag.

The use of a simple POS tagger is motivated by analysis conducted in Sect. 3.5, where a manual evaluation of parse trees indicated that parsers provide good structures over disfluent English by incorrectly tagging words. Assigning to words their most likely POS tags according to unigram estimates should allow a discriminative language model to better identify and penalise reordering mistakes.

4.2.2 Deep features

Sequential rules (*POS, SEQ-B and SEQ-C*) From the full parse tree, we extract three different layers. Fig. 4b shows the three annotation layers we extract from the parse tree shown in Fig. 4a. In Fig. 4b (POS), the sequence comprises the POS tags for each word. Fig. 4b (SEQ-B) captures chunk-based sequences by associating with each word the first non-POS ancestor node. For each word, it is also indicated whether it starts or continues a shallow chunk (^{b} for the former and ^{c} for the latter). The sequence in Fig. 4b (SEQ-C) is similar, but includes the POS tag of each word.

From the POS, SEQ-B and SEQ-C layers, as well as from the S-POS and CRF-POS output, we extract the following features, where w_i is a word at index i, and t is a tag specific to the layer we are extracting from:

$$(t_{i-2}t_{i-1}t_i), (t_{i-1}, t_i), (t_i), (t_i w_i)$$

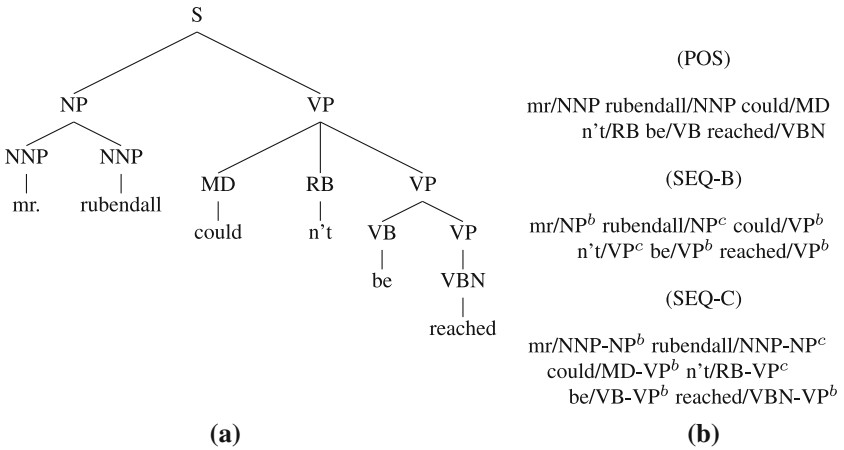


Fig. 4 In **a** we show an example parse tree, and in **b** we show the POS, SEQ-B and SEQ-C sequences extracted from **a**

Context free grammar rules (*CFG*) Each feature is a basic CFG rule used in the parse of a sentence. CFGs form the building blocks of most parsers, and except for rules from pre-terminals to leaf nodes, are entirely non-lexical. Thus these features capture information about categorial reordering decisions, such as Noun Phrase (NP) before Verb Phrase (VP), or vice versa. While simple, this feature type can capture long-range reordering mistakes.

Syntactic head features (*HEAD*) We model head-parent relationships, such as NP headed by NN (syntactic), represented by NP/NN, or NP headed by car, NP/car (lexical). These features are extracted for each non-terminal (NT) in the tree. A head denotes the most important syntactic child of a phrase category. Heads are extracted using the hand crafted rules defined in Appendix A of (Collins 1999).

In addition to the simple head-parent relationships, we also model more complex head-to-head dependencies within the parse tree. Given a parent NT P in a tree and its NT children ($C \cdots C_k$), we model the relationship between P, the head child of $P\,C_h$, and each sibling node of C_h. We denote the relative position between C_h and the sibling C_k with an integer, 1 if adjacent, 2 if not, positive if C_k is to the right, negative if to the left of C_h. Finally we note the lexical or POS head of C_h and C_k. The final feature is of the form: $P, HC, C_k, \{+, -\}, lex/POS, lex/POS$. Examples from Fig. 4a include:

VP,MD,VP,2,could,be
VP,MD,VP,2,could,VBN
VP,MD,VP,2,MD,be
VP,MD,VP,2,MD,VBN

Tree depth (*T-DEPTH*) This feature measures the maximum height of the tree. The intuition is that a deep complex structure is indicative of a particularly disfluent and ungrammatical sentence. Comparing the baseline SMT translations with the reference

sentences for MT04, we see that the SMT output has on average a far higher tree depth; the summed difference is between 20 and 56 levels higher for the SMT translations. We normalise this feature by sentence length.

Unit productions (*UNIT-P*) This feature takes the sum of all unit productions in the associated parse tree. Comparing all the baseline SMT translations with the reference sentences for MT04, we see that the SMT output has a total difference of between 7.5 and 20 more unit productions. It appears that overgeneration of unit productions is indicative of a disfluent sentence, and thus we include it as a feature.

NT count (*NT-C*) This feature counts the number of non-terminal types in a sentence, normalised by sentence length. The aim is to capture the over/underproduction of certain feature types, a common problem in SMT output (e.g, a dropped verb).

Node count (*NO-C*) This feature counts the number of nodes in a parse, normalised by sentence length. Despite being quite a simple feature, we note a general trend for parses of SMT sentences to contain more nodes than human-produced translations; the SMT output for MT04 has a total difference of between 71 and 205 more nodes.

4.2.3 Shallow features

POS n-grams We explore the use of POS n-gram features, from unigram to trigram features.

Verb agreement (*VERBAGR*) The verb agreement feature captures agreement between verb tenses that should match. We extract this feature by starting with each coordinating conjunction and comma in a sentence, and examine a window of 5 words on either side for verbs. If there are multiple verbs in this window, we return the nearest one either side. This feature is extracted only if we find a verb both to the left and right within the context length. For example, given the sentence "George/NNP was/VBD shouting/VBG and/CC screaming/VBG", the verb agreement feature would be:

VBG CC VBG

This feature can discriminate between the correct form "shouting and screaming " and the incorrect "shouting and screamed". Note this is not a trigram POS feature, as the verbs do not have to be adjacent to the comma or coordinating conjunction.

NT length (*POSNUM*) It is also possible to extract features from the POS layers that capture frequency-based information. In particular, we wish to model the frequency of POS types for a given translation length. Features are of the form:

$$length(x)/num(POS, x)$$

The length of a sentence is represented by $length(x)$, and the frequency with which a specific POS tag occurs in the hypothesis translation x is $num(POS, x)$. These features tie the number of POS tags to the length of a sentence, and thus model the under/overproduction of certain POS types for specific sentence lengths. Here, we examine five such types: verbs, nouns, adverbs, adjectives and determiners.

POS absence (*NOPOS*) A similar feature is one that models a lack of certain POS types, regardless of sentence length. Here again we model a lack of either verbs, nouns, adverbs, adjectives or determiners.

5 Experiments

In this section, we evaluate the impact of different syntactic features used by a discriminative language model on the MT04, MT05 and MT06 test sets. Note that we looked at the baseline output of the MT04 test set in motivating features proposed in this article. However, we did not examine the MT05 and MT06 test sets, which remain unseen. The SMT system and settings remain the same as those described in Sect. 3.2.

5.1 Parameter optimisation

The SMT system is optimised on the MT02 and MT03 data sets. Since the parameters of the perceptron reranker also require optimisation, the development set was split into K folds. MERT was run on the union of the *K-1* folds to optimise the parameters. The resulting setting was used to translate the remaining fold and to generate the n-best lists used for learning the optimal parameter settings of the perceptron reranker. The n-best lists contain the top-1000 most likely and distinct translation candidates, as it is possible that different alignments can lead to sentences which are lexically identical but have different derivations. Untranslated source words were not removed from translations. Note that the Moses baseline we compare against was still trained on all the development data in one go.

To optimise the β value in Eq. (1), we performed a grid search, with increments of 0.1 examined between 0 and 1, and increments of 1 at 2^x thereafter, on the MT0203 set.

As we parse SMT output, all sentences were tokenised and lowercased in accordance with the output of the SMT system prior to training the parser. The simple unigram tagger was trained analogously, also on sections 02-21 of the PTB. A sentence was assigned the ⟨NOPARSE⟩ feature if the parser failed to generate a parse for it. In such a situation, excluding the ⟨NOPARSE⟩ feature, a syntactic reranker can only take into account lexical features when assigning a score to such a sentence. Note that the first sentence of each n-best list was always assigned a parse. The tagging accuracy of the parser and two POS taggers are as follows: CRF 97%, CM2 94.4% and S-POS 86.8%.

5.2 Results

Having detailed the different annotation layers and syntactic features we intend to explore, we now present experimental results. Table 4 presents Moses baseline 1-best results on MT04, MT05 and MT06 test sets. In addition to the Moses baseline, we present results using the averaged n-gram reranking model using unigram, bigram and trigram lexical features, as used by Li and Khudanpur (2008). Finally, we also

Table 4 Moses baseline, n-gram-reranked and oracle results on MT04, MT05 and MT06

	MT04	MT05	MT06
Moses	48.97	53.92	38.40
+ DLM n-gram	49.57	54.42	39.08
Oracle	61.09	66.34	50.11

Table 5 Results on MT development and test sets using syntactic features from full parse trees

	MT0203	MT04	MT05	MT06
Moses	51.27	48.97	53.92	38.40
+ DLM n-gram	59.87	49.57	54.42	39.08
+ DLM n-gram + POS	59.70	49.47	**54.48**	39.07
+ DLM n-gram + SEQ-B	58.52	49.09	54.11	39.47
+ DLM n-gram + SEQ-C	60.37	49.46	54.19	39.07
+ DLM n-gram + CFG	59.89	49.53	54.44	39.58
+ DLM n-gram + HEAD	**61.53**	49.44	54.09	33.45
+ DLM n-gram + MAX-D	58.79	49.42	54.08	39.61
+ DLM n-gram + UNIT-P	59.51	**49.81**	54.39	**39.76**
+ DLM n-gram + NT-C	58.82	49.51	54.20	39.68
+ DLM n-gram + NO-C	53.52	47.14	52.68	36.92

We highlight in bold the largest scores

present oracle results in the last row of Table 4, demonstrating the large room left for improvement.

5.2.1 Deep features

In Table 5 we present the results of using our perceptron rerankers with features extracted from full parse trees. The use of features from full parse trees did not help at all for MT04, apart from the UNIT-P model, which gave improvements of 0.34 BLEU. For MT05, the CFG and POS feature sets show only small improvements. Note for MT05 the UNIT-P model no longer gives improvements above the n-gram only model. For the MT06 test set, all syntactic models apart from HEAD achieve improvements. These improvements against the lexical-only reranker do not hold for MT04 and MT05. Robustness is a problem; given unseen test data, we do not know whether the inclusion of syntactic features from full parse trees will improve or harm the translation quality of the system.

5.2.2 POS layers compared

In comparing the effect of different syntactic toolkits, we conduct experiments with features extracted from the POS taggers. The results are displayed in Table 6.

The CRF DLM outperforms the n-gram only DLM model on all three test sets. The S-POS DLM yields gains over the DLM n-gram model on all three of the test sets also. Even though our S-POS tagger uses no back-off model or context, for two of the

Table 6 BLEU scores and improvements when using features from our two POS taggers and POS annotations from the full tree parser

	MT04	MT05	MT06
DLM n-gram	49.57	54.42	39.08
DLM n-gram + POS	49.47	54.48	39.07
Improvement	−0.10	0.06	−0.01
DLM n-gram + CRF	**49.74**	54.51	39.45
Improvement	**0.17**	0.09	0.37
DLM n-gram + S-POS	49.59	**54.60**	**39.48**
Improvement	0.02	**0.18**	**0.40**

POS features extracted from a simple unigram, maximum likelihood tagger give the largest improvements on two of the three sets
We highlight in bold the best scores

Table 7 Model results using POS tag frequency (vn, dn and allnum), lack of POS type (noall) and verb agreement (verbagr) features

	MT04	MT05	MT06
Moses	48.97	53.92	38.40
+ DLM n-gram	49.57	54.42	39.08
++ S-POS+V+DNUM	49.65†	**54.60**‡	39.67‡
++ S-POS+ALLNUM	49.65	**54.60**	39.67‡
++ S-POS+NOALL	**49.70**‡	54.46	**39.69**‡
++ S-POS+VERBAGRE	49.44	54.56	39.55‡

We conduct significance tests between the syntactic models and the DLM n-gram model. Significance at ‡ $p < 0.01$. Significance at † $p < 0.05$
We highlight in bold the best scores

three test sets, it provides larger gains than the CRF tagger. Because the S-POS tagger results in higher scores than the CRF tagger for two of the three test sets, we only use the simple POS annotation layer for the following experiments.

5.2.3 Shallow features using simple POS tagger

Table 7 summarizes the results of using the POSNUM, NOPOS and VERBAGR features. As for the POSNUM and NOPOS features, we look at specific POS categories, with POS replaced by the respective type V (verb), N (noun), D (determiner), RB (adverb), JJ (adjective) and ALL (all of the previous five types). For MT04, the best-performing model is S-POS+noall, with a significant improvement at $p < 0.01$ over the DLM n-gram model of 0.13 corpus-level BLEU points.[3] For MT05, the best-performing model is S-POS+V+DNUM with a significant improvement of 0.18 BLEU points at $p < 0.01$. The S-POS+ALLNUM model gives the same

[3] Statistical significance is calculated using the paired bootstrap resampling method (Koehn 2004).

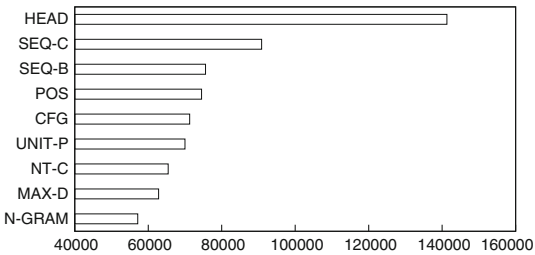


Fig. 5 Number of active features (all features with non-zero weights) in each model

absolute BLEU improvement for MT05, but is insignificant. For MT06, we have a larger improvement of 0.41 BLEU, again at $p < 0.01$, using S-POS+NOALL. The S-POS+V+DNUM model is not the best-performing model on MT04 or MT06, but consistently gives significant improvements.

5.2.4 Deep + shallow feature combinations

Finally, we investigate whether a combination of deep and shallow features leads to improvements. As it is infeasible to try all feature combinations together, we pick the best-performing deep feature type UNIT-P, and combine this with features extracted from our S-POS tagger. The combination of deep and shallow feature types does not lead to improvements over those presented in Table 7, so we conclude that the deep features examined are redundant.

6 Discussion

An explanation for the underperformance of features derived from full parse trees is overtraining. Examining the performance of the syntactic models on the MT0203 development set, SEQ-C and HEAD models perform considerably better than the n-gram-only model and other syntactic models. This performance does not carry through to the test sets, indicating overtraining. Looking at the number of active (non-zero-weighted) features contained in the syntactic models in Fig. 5, we see that adding syntactic features to our model increases the model size, and also that models SEQ-C and HEAD have the most features. We posit that these models are learning features that explain the development data well, but do not generalise to unseen data.

Overfitting is exacerbated by the lack of parses from which to extract parses. This is because we do not apply any sentence-level thresholding, meaning we are unable to parse every sentence in all the n-best lists, although every n-best list contained at least one parse. For the MT0203 training set, only 87.3% of the sentences had a parse. This means that the large feature spaces demonstrated by Fig. 5 were coming from a reduced example set. For the test sets, only between 80.7 and 82.6% of the sentences had a parse, thus limiting the ability of the syntactic features to help improve translation quality. The combination of a large feature space, over fewer training samples, leads to poor performance when using sparse features extracted from parsers.

Table 8 N-gram precision rates and relative improvements on MT test sets above the Moses baseline and n-gram reranker

Test set	System	n-gram precision (%)			
		1	2	3	4
MT04	Moses	81.46	57.80	41.17	29.70
	+n-gram	**81.86**	58.36	41.72	30.28
	++syntax	81.76	**58.48**	**41.92**	**30.43**
Improvement (%)		0.4/−0.1	1.2/0.2	1.8/0.5	2.5/0.5
MT05	Moses	83.18	62.34	46.66	34.93
	+n-gram	**83.31**	62.74	47.20	35.54
	++syntax	83.28	**62.96**	**47.43**	**35.74**
Improvement (%)		0.1/−0.04	1/0.3	1.7/0.5	2.3/0.6
MT06	Moses	74.17	47.45	31.27	21.05
	+n-gram	**74.43**	47.84	31.75	21.50
	++syntax	74.31	**47.92**	**31.87**	**21.58**
Improvement (%)		0.2/−0.2	1/0.2	1.9/0.4	2.5/0.4

We highlight in bold the highest precision scores, and show percentage improvements of our syntactic model above the Moses baseline and lexical-only reranker. For bigram, trigram and four-gram precision, syntactic rerankers achieve the highest scores

Table 8 presents the individual n-gram precision rates for our best syntactic models in comparison to the n-gram only DLM. There is a degradation in relative unigram precision on all three sets, but we see an increase in bigram, trigram and four-gram precision, indicating that our syntactic features resolve some word reordering problems.

To see how the S-POS features help, Table 9 presents the different POS sequences assigned by the three different syntactic tools to the translation: *he reiterated " full support of the islamic republic for islamic government interim " in afghanistan*. This sentence is chosen by the perceptron reranker using POS features from the CM2 parser. The bigram "government interim" is tagged as "NN NN" by CM2 and the CRF tagger. This feature has a positive weight, and thus is not penalised. Only S-POS tags "interim" as an adjective. In the last row of Table 9 we show the translation chosen by the perceptron reranker using features from the S-POS tagger. This leads to an improvement of 17.34 sentence-level BLEU points, and is an example of the significant gains we make using our S-POS tagger.

7 Conclusion

We have examined the ability of a state-of-the-art parser to discriminate between fluent and disfluent English, finding that it outperforms a standard four-gram language model. Encouraged by this, we have proposed to use a syntactic discriminative language model to rerank translation hypothesis. The main contributions of this article are the extension of lexical discriminative rerankers with syntactic features, and the

Table 9 POS assignments made by the three syntactic tools for the sentence *he reiterated "full support of the islamic republic for islamic government interim" in afghanistan*

System	POS sequence	BLEU
CM2	PRP/he VBD/reiterated "/" JJ/full NN/support IN/of DT/the NNP/islamic NNP/republic IN/for JJ/islamic **NN/government NN/interim** IN/in "/" NNP/afghanistan./.	50.30
CRF	he/PRP reiterated/VBD "/" full/JJ support/NN of/IN the/DT islamic/JJ republic/NN for/IN islamic/JJ **government/NN interim/NN** "/" in/IN afghanistan/JJ./.	50.30
S-POS	PRP/he VBD/reiterated "/" JJ/full NN/support IN/of DT/the NNP/islamic NNP/republic IN/for NNP/islamic **NN/government JJ/interim** "/" IN/in NNP/afghanistan./.	50.30
S-POS	PRP/he VBD/reiterated "/" JJ/full NN/support IN/of DT/the NNP/islamic NNP/republic IN/for NNP/islamic **JJ/interim NN/government** IN/in NNP/afghanistan./. "/"	**67.64**

We present sentence-level BLEU scores for both translations

study of the use of a simple, non-context-aware POS tagger that overcomes problems encountered when using standard syntactic toolkits. Extensive experiments using our syntactic models demonstrate significant improvements in BLEU over non-reranked output and lexical-only reranked models. Furthermore, we have conducted experiments examining the utility of deep and shallow syntactic annotation layers, and the different features extractable from them.

We show that deep features, which are used with the intention of making more generalisable models, do not help within the perceptron reranking setting as they overfit the training data, leading to problems with robustness of results over different test sets. As far as future work is concerned, we believe an examination of the use of partial parsers may lead to a successful bridge between the benefits of deep features from a full parser and the coverage of data points from POS and other shallow tools.

Acknowledgements The authors would like to thank Valentin Jijkoun, Sophia Katrenko and the anonymous reviewers for their insightful comments and helpful discussions. This work has been funded in part by the European Commission through the CoSyne project FP7-ICT-4-248531.

References

Arun A, Koehn P (2007) Online learning methods for discriminative training of phrase based statistical machine translation. In: Machine translation summit XI: proceedings, Copenhagen, pp 15–20

Bikel DM (2002) Design of a multi-lingual, parallel-processing statistical parsing engine. In: HLT 2002: human language technology conference, proceedings of the second international conference on human language technology research, San Diego, pp 178–182

Bilmes JA, Kirchhoff K (2003) Factored language models and generalized parallel backoff. In: HLT-NAACL 2003: conference combining human language technology conference series and the North American chapter of the Association for Computational Linguistics conference series, Edmonton, pp 4–6

Birch A, Osborne M, Koehn P (2007) CCG supertags in factored statistical machine translation. In: Proceedings of the second workshop on statistical machine translation (WMT 2007), Prague, pp 9–16
Blunsom P, Cohn T, Osborne M (2008) A discriminative latent variable model for statistical machine translation. In: ACL-08: HLT, 46th annual meeting of the Association for Computational Linguistics: human language technologies, proceedings of the conference, Columbus, pp 200–208
Brown PF, Pietra VJ, de Souza PV, Lai JC, Mercer RL (1992) Class-based n-gram models of natural language. Comput Linguist 18(4):467–479
Callison-Burch C, Osborne M, Koehn P (2006) Re-evaluating the role of BLEU in machine translation research. In: EACL-2006: 11th conference of the European chapter of the Association for Computational Linguistics, Proceedings of the conference, Trento, pp 249–256
Carter S, Monz C (2009) Parsing statistical machine translation output. In: Proceedings of the language & technology conference (LTC 2009), Poznań, pp 270–274
Carter S, Monz C (2010) Discriminative syntactic reranking for statistical machine translation. In: AMTA 2010: proceedings of the ninth conference of the Association for Machine Translation in the Americas, Denver, pp 3–12
Chang PC, Toutanova K (2007) A discriminative syntactic word order model for machine translation. In: proceedings of the 45th annual meeting of the Association for Computational Linguistics (ACL 2007), Prague, pp 9–16
Chen SF, Goodman J (1998) An empirical study of smoothing methods for language modelling. Tech. Rep. TR-10-98, University of Harvard, Cambridge
Chen X, Wang H, Lin X (2009) Learning to rank with a novel kernel perceptron method. In: Proceedings of the 18th ACM conference on information and knowledge management (CIKM 2009), Hong Kong, pp 505–512
Chiang D (2005) A hierarchical phrase-based model for statistical machine translation. In: 43rd annual meeting of the Association for Computational Linguistics (ACL 2005), Ann Arbor, pp 263–270
Chiang D (2007) Hierarchical phrase-based translation. Comput Linguist 33(2):201–228
Chiang D, Marton Y, Resnik P (2008) Online large-margin training of syntactic and structural translation features. In: EMNLP 2008: 2008 conference on empirical methods in natural language processing, Proceedings of the conference, Honolulu, pp 224–233
Chiang D, Wang W, Knight K (2009) 11,001 new features for statistical machine translations. In: Human language technologies: the 2009 annual conference of the North American chapter of the Association for Computational Linguistics, proceedings of the conference, Boulder, pp 218–226
Collins M (1997) Three generative, lexicalized models for statistical parsing. In: Cohen PR, Wahlster W (eds) 35th annual meeting of the Association for Computational Linguistics and 8th conference of the European chapter of the Association for Computational Linguistics, proceedings of the conference, Madrid, pp. 16–23
Collins M (1999) Head-driven statistical models for natural language parsing. PhD thesis, University of Pennsylvania, Philadelphia, Pennsylvania
Collins M, Duffy N (2002) New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In: 40th annual meeting of the Association for Computational Linguistics, proceedings of the conference, Philadelphia, pp 263–270
Collins M, Roark B, Saraclar M (2005) Discriminative syntactic language modeling for speech recognition. In: 43rd annual meeting of the Association for Computational Linguistics (ACL 2005), Ann Arbor, pp 507–514
Crammer K, Singer Y (2001) Pranking with ranking. In: Proceedings of the twenty-fifth annual conference on advances in neural information processing systems (NIPS 2001), Vancouver, pp 641–647
Elsas JL, Carvalho VR, Carbonell JG (2008) Fast learning of document ranking functions with the committee perceptron. In: Proceedings of the international conference on web search and web data mining (WSDM 2008), Stanford, pp 55–64
Emami A, Papineni K, Sorensen J (2007) Large-scale distributed language modeling. In: Proceedings of the international conference on acoustics, speech and signal processing (ICASSP 2007), Honolulu, pp 37–40
Freund Y, Schapire RE (1999) Large margin classification using the perceptron algorithm. Mach Learn 37(3):277–296
Gallant SI (1999) Perceptron based learning algorithms. IEEE Trans Neural Netw 1(2):179–191

Hasan S, Bender O, Ney H (2006) Reranking translation hypotheses using structural properties. In: EACL-2006: 11th conference of the European chapter of the Association for Computational Linguistics, proceedings of the conference, Trento, pp 41–48

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 conference on empirical methods in natural language processing, Barcelona, pp 388–395

Koehn P, Hoang H (2007) Factored translation models. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CONLL 2007), Prague, pp 868–876

Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. In: HLT-NAACL 2003: conference combining human language technology conference series and the North American chapter of the Association for Computational Linguistics conference series, Edmonton, pp 48–54

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: ACL 2007, proceedings of the interactive poster and demonstration sessions, Prague, pp 177–180

Kulesza A, Shieber, S (2004) A learning approach to improving sentence-level MT evaluation. In: TMI-2004: proceedings of the tenth conference on theoretical and methodological issues in machine translation, Baltimore, pp 75–84

Li Z, Khudanpur S (2008) Large-scale discriminative n-gram language models for statistical machine translation. In: AMTA-2008: MT at work: proceedings of the eighth conference of the Association for Machine Translation in the Americas, Waikiki, pp 133–142

Liang P, Bouchard-Côté A, Klein D, Taskar B (2006) An end-to-end discriminative approach to machine translation. In: COLING ACL 2006, 21st international conference on computational linguistics and 44th annual meeting of the Association for Computational Linguistics, proceedings of the conference, Sydney, pp 761–768

Lin CY, Och FJ (2004) Orange: a method for evaluating automatic evaluation metrics for machine translation. In: 20th international conference on computational linguistics, proceedings, vol I, Geneva, pp 501–507

Marcus M, Kim G, Marcinkiewicz MA, Macintyre R, Bies A, Ferguson M, Katz K, Schasberger B (1994) The Penn Treebank: annotating predicate argument structure. In: Human language technology, proceedings of a workshop, Plainsboro, pp 114–119

McDonald R (2007) Characterizing the errors of data-driven dependency parsing models. In: EMNLP-CoNLL 2007: proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, Prague, pp 121–131

Mohit B, Hwa R (2007) Localization of difficult-to-translate phrases. In: Proceedings of the second workshop on statistical machine translation (WMT 2007), Prague, pp 248–255

Och FJ (2003) Minimum error rate training in statistical machine translation. In: 41st annual meeting of the Association for Computational Linguistics, proceedings of the conference, Sapporo, pp 160–167

Och FJ, Ney H (2000) Improved statistical alignment models. In: 38th annual meeting of the Association for Computational Linguistics, proceedings of the conference, Hong Kong, pp 440–447

Och FJ, Gildea D, Khudanpur S, Sarkar A, Yamada K, Fraser A, Kumar S, Shen L, Smith D, Eng K, Jain V, Jin Z, Radev D (2003) Syntax for statistical machine translation. Tech. Rep. IRCS-00-07, Johns Hopkins 2003 Summer Workshop, Baltimore

Och FJ, Gildea D, Khudanpur S, Sarkar A, Yamada K, Fraser A, Kumar S, Shen L, Smith D, Eng K, Jain V, Jin Z, Radev D (2004) A smorgasbord of features for statistical machine translation. In: HLT-NAACL 2004: human language technology conference of the North American chapter of the Association for Computational Linguistics, proceedings of the main conference, Boston, pp 161–168

Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: a method for automatic evaluation of machine translation. In: 40th annual meeting of the Association for Computational Linguistics, proceedings of the conference, Philadelphia, pp 311–318

Post M, Gildea D (2008) Parsers as language models for statistical machine translation. In: AMTA-2008: MT at work: proceedings of the Eighth conference of the Association for Machine Translation in the Americas, Waikiki, pp 172–181

Roark B, Saraclar M, Collins M (2004a) Corrective language modeling for large vocabulary ASR with the perceptron algorithms. In: Proceedings of the international conference on acoustics, speech and signal processing (ICASSP 2004), Montreal, pp 749–752

Roark B, Saraclar M, Collins M, Johnson M (2004b) Discriminative language modeling with conditional random fields and the perceptron algorithm. In: ACL-04, 42nd annual meeting of the Association for Computational Linguistics, proceedings of the conference, Barcelona, pp 47–54

Roark B, Saraclar M, Collins M (2007) Discriminative n-gram language modeling. Comput Speech Lang 21(2):373–392

Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Neurocomput Found Res 65(6):386–408

Shen L, Sarkar A, Och FJ (2004) Discriminative reranking for machine translation. In: HLT-NAACL 2004: human language technology conference of the North American chapter of the Association for Computational Linguistics, proceedings of the main Conference, Boston, pp 177–184

Singh-Miller N, Collins C (2007) Trigger-based language modeling using a loss-sensitive perceptron algorithm. In: Proceedings of the international conference on acoustics, speech and signal processing (ICASSP 2007), Honolulu, pp 25–28

Stolcke A (2002) SRILM—an extensible language modeling toolkit. In: Proceedings of the international conference on spoken language processing (ICSLP 2002), Denver, pp 901–904

Tillmann C, Zhang T (2006) A discriminative global training algorithm for statistical MT. In: COLING ACL 2006, 21st international conference on computational linguistics and 44th annual meeting of the Association for Computational Linguistics, proceedings of the conference, Sydney, pp 721–728

Watanabe T, Suzuki J, Tsukada J, Isozaki H (2007) Online large-margin training for statistical machine translation. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CONLL 2007), Prague, pp 764–773