Modelling and Optimizing on Syntactic N-Grams for Statistical Machine Translation

Modelling and Optimizing on Syntactic N-Grams for Statistical Machine Translation

Rico Sennrich
School of Informatics
University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB
Scotland, UK

Abstract

The role of language models in SMT is to promote fluent translation output, but traditional n-gram language models are unable to capture fluency phenomena between distant words, such as some morphological agreement phenomena, subcategorisation, and syntactic collocations with string-level gaps. Syntactic language models have the potential to fill this modelling gap. We propose a language model for dependency structures that is relational rather than configurational and thus particularly suited for languages with a (relatively) free word order. It is trainable with Neural Networks, and not only improves over standard n-gram language models, but also outperforms related syntactic language models. We empirically demonstrate its effectiveness in terms of perplexity and as a feature function in string-to-tree SMT from English to German and Russian. We also show that using a syntactic evaluation metric to tune the log-linear parameters of an SMT system further increases translation quality when coupled with a syntactic language model.

1 Introduction

Many languages exhibit fluency phenomena that are discontinuous in the surface string, and are thus not modelled well by traditional n-gram language models. Examples include morphological agreement, e.g. subject-verb agreement in languages that do not (exclusively) follow SVO word order, subcategorisation, and collocations involving distant, but syntactically linked words.

Syntactic language models try to overcome the limitation to a local n-gram context by using syntactically related words (and non-terminals) as context information. Despite their theoretical attractiveness, it has proven difficult to improve SMT with parsers as language models (Och et al., 2004; Post and Gildea, 2008).

This paper describes an effective method to model, train, decode with, and weight a syntactic language model for SMT. While all these aspects are important for successfully applying a syntactic language model, our primary contributions are a novel dependency language model which improves over prior work by making relational modelling assumptions, which we argue are better suited for languages with a (relatively) free word order, and the use of a syntactic evaluation metric for optimizing the log-linear parameters of the SMT model.

While language models that operate on words linked through a dependency chain – called *syntactic n-grams* (Sidorov et al., 2013) – can improve translation, some of the improvement is invisible to an n-gram metric such as BLEU. As a result, tuning to BLEU does not show the full value of a syntactic language model. What does show its value is an optimization metric that operates on the same syntactic n-grams that are modelled by the dependency LM.

The paper is structured as follows. Section 2 describes our relational dependency language model; section 3 describes our neural network training procedure, and the integration of the model into an SMT decoder. We describe the syntactic evaluation metric we use for tuning in Section 4. The language models are evaluated on the basis of perplexity and SMT

Transactions of the Association for Computational Linguistics, vol. 3, pp. 169–182, 2015. Action Editor: Philipp Koehn.
Submission batch: 11/2014; Revision batch 2/2015; Published 3/2015.

performance in section 5. We discuss related work in section 6, and finish with concluding remarks in section 7.

2 A Relational Dependency Language Model

As motivation, and working example for the model description, consider the dependency tree in Figure 1, which is taken from the output of our baseline string-to-tree SMT system.[1] The output contains two errors:

- a morphological agreement error between the subject *Ergebnisse* (plural) and the finite verb *wird* (singular).
- a subcategorisation error: *überraschen* is transitive, but the translation has a prepositional phrase instead of an object.

While these errors might not have occurred if the words involved were adjacent to one another here and throughout the training set, non-adjacency is common, especially where the distance between subject and finite verb, or between a full verb and its arguments can be arbitrarily long.

Prior work on syntactic language modelling has typically focused on English, and we argue that some modelling decisions do not transfer well to other languages. The dependency models proposed by Shen et al. (2010) and Zhang (2009) rely heavily on structural information such as the direction and distance of the dependent from the parent. In a language where the order of syntactic dependents is more flexible than in English, such as German[2], grammatical function (and thus the inflection) is hard to predict from the dependent order. Instead, we make dependency labels, which encode grammatical relations, a core element of our model.[3]

Shen et al. (2010) propose a model that estimates probability of each token given its parent and/or preceding siblings. We start with a variant of their model that does not hard-code configurational modelling assumptions, and then extend it by including dependency labels.

2.1 Unlabelled Model

Let S be a sequence of terminal symbols $w_1, w_2, ..., w_n$ with a dependency topology T, and let $h_s(i)$ and $h_a(i)$ be lists of heads of preceding siblings and ancestors of w_i according to T, from closest to furthest. In our example in Figure 1:

- $w_4 =$ *jüngsten*
- $h_s(4) = (\textit{der})$
- $h_a(4) = (\textit{Umfrage}, \textit{Ergebnisse}, \textit{wird}, \epsilon)$

Note that h_a and its subsequences are instances of syntactic n-grams. For this model, we follow related work and assume that T is available (Popel and Marecek, 2010), approximating $P(S)$ as $P(S|T)$. We make the Markov assumption that the probability of each word only depends on its preceding siblings[4] and ancestors, and decompose the probability of a sentence like this:

$$\begin{aligned} P(S) &= P(w_1, w_2, ..., w_n) \\ &\approx \prod_{i=1}^{n} P(w_i | h_s(i), h_a(i)) \end{aligned} \quad (1)$$

We further make the Markov assumption that only a fixed window of the closest q siblings, and the closest r ancestors, affect the probability of a word.

$$P(S) \approx \prod_{i=1}^{n} P(w_i | h_s(i)_1^q, h_a(i)_1^r) \quad (2)$$

Equation 2 represents our basic, unlabelled model. It differs from that of Shen et al. (2010) in two ways.

[1] The tree is converted into constituency format for compatibility with SCFG decoding algorithms, with dependency edges represented as non-terminal nodes.

[2] German has a strict word order within noun phrases and for the placement of verbs, but has different word order for main clauses and subordinated clauses, and some flexibility in the order of dependents of a verb.

[3] Tsarfaty (2010) classifies parsing approaches into *configurational* approaches that rely on structural information, and *relational* ones that take grammatical relations as primitives. While she uses dependency syntax as a prototypical example of relational approaches, the dependency LM by Shen et al. (2010) would fall into the configurational category, while ours is relational.

[4] Shen et al. (2010) use the siblings that are between the word and its parent, i.e. the following siblings if the word comes before its parent. We believe both preceding and following siblings are potentially useful, but leave expansion of the context to future work.

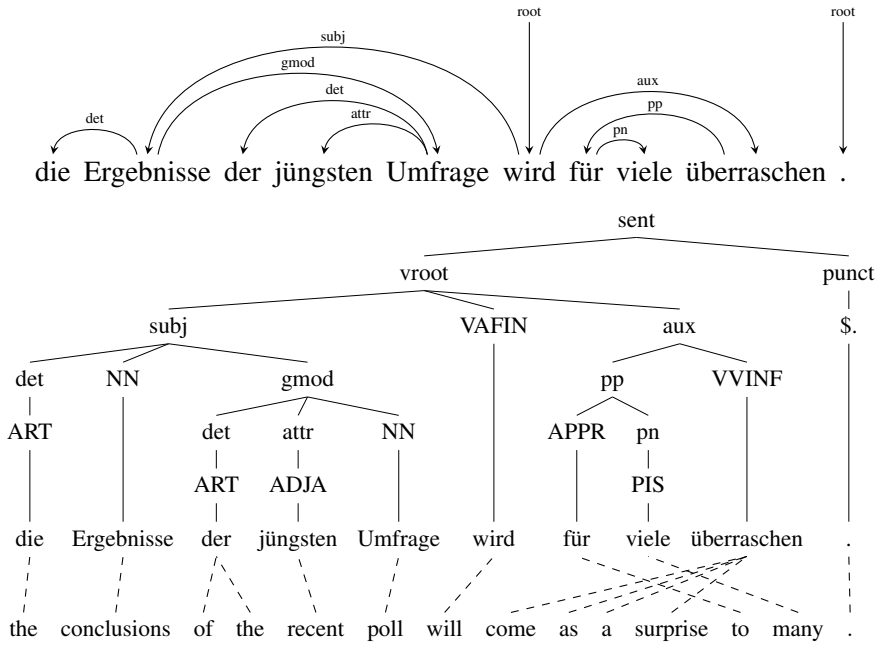

Figure 1: Translation output of baseline English→German string-to-tree SMT system with original dependency representation and conversion into constituency representation.

First, it uses separate context windows for siblings and ancestors. In contrast, Shen et al. (2010) treat the ancestor as the first symbol in a context window that is shared between the ancestor and siblings. Our formulation encodes our belief that the model should always assume dependence on the r nearest ancestor nodes, regardless of the number of siblings. Secondly, Shen et al. (2010) separate dependents to the left and to the right of the parent. While the fixed SVO verb order in English is compatible with such a separation, allowing P_L to model subjects, P_R to model objects, most arguments can occur before or after the head verb in German main clauses. We thus argue that left and right dependents should be modelled by a single model to allow for sharing of statistical strength.[5]

2.2 Labelled Model

The motivation for the inclusion of dependency labels is twofold. Firstly, having dependency labels in the context serves as a strong signal for the prediction of the correct inflectional form. Secondly, dependency labels are the appropriate level of abstraction to model subcategorisation frames.

Let D be a sequence of dependency labels $l_1, l_2, ..., l_n$, with each label l_i being the label of the incoming arc at position i in T, and $l_s(i)$ and $l_a(i)$ the list of dependency labels of the siblings and ancestors of w_i, respectively. Continuing the example for w_4, these are:

- $l_4 = \textit{attr}$
- $l_s(4) = (\textit{det})$
- $l_a(4) = (\textit{gmod}, \textit{subj}, \textit{vroot}, \textit{sent})$

We predict both the terminal symbols S and dependency labels D. The latter lets us model subcategorisation by penalizing unlikely relations, e.g. objects whose parent is an intransitive verb. We decompose $P(S, D)$ into $P(D) \times P(S|D)$ to obtain:

$$\begin{aligned} P(S,D) &= P(D) \times P(S|D) \\ &\approx \prod_{i=1}^{n} P_l(i) \times P_w(i) \\ P_l(i) &= P(l_i | h_s(i)_1^q, l_s(i)_1^q, h_a(i)_1^r, l_a(i)_1^r) \\ P_w(i) &= P(w_i | h_s(i)_1^q, l_s(i)_1^q, h_a(i)_1^r, l_a(i)_1^r, l_i) \end{aligned} \tag{3}$$

[5] Similar arguments have been made for parsing of (relatively) free word-order languages, e.g. by Tsarfaty et al. (2009).

2.3 Head and Label Extraction

We here discuss some details for the extraction of the context h_s and h_a. Dependency structures require no language-specific head extraction rules, even in a converted constituency representation. In the constituency representation shown in Figure 1, each non-terminal node in the tree that is not a pre-terminal has exactly one pre-terminal child. The head of a non-terminal node can thus be extracted by identifying the pre-terminal child, and taking its terminal symbol as head. An exception is the virtual node $sent$, which is added to the root of the tree to combine subtrees that are not connected in the original grammar, e.g. the main tree and the punctuation symbol. If a node has no pre-terminal child, we use a special token ϵ as its head.

If the sibling of a node is a pre-terminal node, we represent this through a special token in h_s and l_s. We also use special out-of-bound tokens (separate for h_s, h_a, l_s and l_a) to fill up the context window if the window is larger than the number of siblings and/or ancestors.

The context extraction rules are language-independent and can be applied to any dependency structure. Language-specific or grammar-specific rules are possible in principle. For instance, for verbal heads in German, one could consider separable verb prefixes part of the head, and thus model differences in subcategorisation between *schlagen* (Engl. *beat*) and *schlagen ... vor* (Engl. *suggest*).

2.4 Predicting the Tree Topology

The model in equation 3 still assumes the topology of the dependency tree to be given, and we remedy this by also predicting pre-terminal nodes, and a virtual $STOP$ node as the last child of each node. This models the position of the head in a subtree (through the prediction of pre-terminal nodes), and the probability that a word has no more dependents (by assigning probability mass to the $STOP$ node).

Instead of generating all n terminal symbols as in equation 3, we generate all m nodes in the dependency tree in top-down, depth-first order, with l_i being PT for pre-terminals, and the node label otherwise, and w_i being either the head of the node, or ϵ if the node has no pre-terminal child. Our final model is given in equation 4.

N	3	4	5
D	det	attr	gmod
S	der	jüngsten	Umfrage
T	5	5	2

N	8	9	10	11	12	13	14	15	16
D	gmod	det	*PT*	*STOP*	attr	*PT*	*STOP*	*PT*	*STOP*
S	Umfrage	der	ϵ	ϵ	jüngsten	ϵ	ϵ	ϵ	ϵ
T	3	8	9	9	8	12	12	8	8

Figure 2: Snippet of prediction steps when generating terminals (top) or all nodes in tree (bottom) for dependency tree in Figure 1.

$$P(S,D,T) \approx \prod_{i=1}^{m} \begin{cases} P_l(i) \times P_w(i), & \text{if } w_i \neq \epsilon \\ P_l(i), & \text{otherwise} \end{cases} \quad (4)$$

Figure 2 illustrates the prediction of a subtree of the dependency tree in Figure 1. Note that T is encoded implicitly, and can be retrieved from D through a stack to which all nodes (except for pre-terminal and $STOP$ nodes) are pushed after prediction, and from which the last node is popped when predicting a $STOP$ node.

3 Neural Network Training and SMT Decoding

We extract all training instances from automatically parsed training text, and perform training with a standard feed-forward neural network (Bengio et al., 2003), using the NPLM toolkit (Vaswani et al., 2013). Back-off smoothing schemes are unsatisfactory because it is unclear which part of the context should be forgotten first, and neural networks elegantly solve this problem. We use two separate networks, one for P_w and one for P_l. Both networks share the same input vocabulary, but are trained and applied independently. The model input is a $(2q + 2r)$-word context vector (+1 for P_w to encode l_i), each word being mapped to a shared embedding layer. We use a single hidden layer with rectified-linear activation function, and noise-contrastive estimation (NCE).

We integrate our dependency language models into a string-to-tree SMT system as additional feature functions that score each translation hypothesis. The model in equation 4 predicts $P(S, D, T)$.

model	input	input	entropy rate
5-gram	5-gram	A B C D E	5.25
bigram	bigram	D E	5.96
5-gram	bigram	ϵ_1 ϵ_2 ϵ_3 D E	6.13

Table 1: Handling unavailable input words by replacing them with null words.

Obtaining the probability of the translation hypothesis $P(S)$ would require the (costly) marginalization over all sequences of dependency labels D and topologies T, but like the SMT decoder itself, we approximate the search for the best translation by searching for the highest-scoring derivation, meaning that we directly integrate P_w and P_l as two features into the log-linear SMT model. We use self-normalized neural networks with precomputation of the hidden layer, which makes the integration into decoding reasonably fast.

The decoder builds the translation bottom-up, and the full context is not available for all symbols in the hypothesis. Vaswani et al. (2013) propose to use a special null word for unavailable context, their embedding being the weighted average of the input embeddings of all other words. We adopt this strategy, with the difference that we use separate null words for each position in the context window in order to reflect distributional differences between the different positions, e.g. between ancestor labels and sibling labels. Symbols are re-scored as more context becomes available in decoding, but poor approximations could affect pruning and thus lead to search errors. In Table 1, we illustrate the use of null words with a 5-gram and a bigram NNLM model. We observe a small increase in entropy when querying the 5-gram model with bigrams, compared to querying a bigram model directly.

Some hierarchical SMT systems allow glue rules which concatenate two subtrees. Since the resulting glue structures do not occur in the training data, we do not estimate their probability in our model. When encountering the root of a glue rule in our language model, we recursively evaluate its children, but ignore the glue node itself. This could introduce a bias towards using more glue rules during translation. To counter this, and encourage the production of linguistically plausible trees, we assign a fixed, high cost to glue rules. Glue rules thus play a small role in our systems, with about 100 glue rule applications per 3000 sentences, and could be abandoned entirely.[6]

4 Optimizing Syntactic N-grams

N-gram based metrics such as BLEU (Papineni et al., 2002) are still predominantly used to optimize the log-linear parameters of SMT systems, and (to a lesser extent) to evaluate the final translation systems. However, n-gram metrics are not well suited to measure fluency phenomena with string-level gaps, and there is a danger that BLEU underestimates the modelling power of dependency language models, resulting in a suboptimal assignment of log-linear weights. As an alternative metric that operates on the level of syntactic n-grams, we use a variant of the head-word chain metric (HWCM) (Liu and Gildea, 2005).

HWCM is a precision metric similar to BLEU, but instead of counting n-gram matches between the translation output and the reference, it compares head-word chains, or syntactic n-grams. HWCM is not only suitable for our task because it operates on the same structures as the dependency language models, but also because our string-to-tree SMT architecture produces trees that can be evaluated directly, without requiring a separate parse of the translation output, a task for which few parsers are optimized. For extracting syntactic n-grams from the reference translations of the respective development and test sets, we automatically parse them, using the same preprocessing as for training.

We count syntactic n-grams of sizes 1 to 4, mirroring the typical usage of BLEU. Banerjee and Lavie (2005) have demonstrated the importance of recall in MT evaluation, and we compute the harmonic mean of precision and recall, which we denote HWCM$_f$, instead of the original, precision-based metric.

5 Evaluation

We perform three evaluations of our dependency language models. Our perplexity evaluation measures model perplexity on the 1-best output of a

[6]For efficiency reasons, our experimental systems only perform SCFG parsing for spans of up to 50 words, and use glue rules to concatenate partial derivations in longer sentences. Better decoding algorithms have reduced the need for this limit (Sennrich, 2014).

baseline SMT system and a human reference translation. Our SMT evaluation integrates the model as a feature function in a string-to-tree SMT system and evaluates its impact on translation quality. Finally, we quantify the effect of different language models on grammaticality by measuring the number of agreement errors of our SMT systems.

We refer to the unlabelled variant of our model (equation 2) as DLM, and to the labelled variant (equation 4) as RDLM, emphasizing that the latter is a *relational* dependency LM.

5.1 Data and Methods

We perform our experiments on English→German data from the WMT 2014 shared translation task (Bojar et al., 2014), consisting of about 4.5 million sentence pairs of parallel data and 120 million sentences of monolingual German data. We train all language models on the German side of the parallel text and the monolingual data. We also perform some experiments on the English→Russian data from the same translation task, with 2 million sentence pairs of parallel data and 34 million sentences of monolingual Russian data.

For a 5-gram Neural Network LM baseline (NNLM), and the dependency language models, we train feed-forward Neural Network language models with the NPLM toolkit. We use 150 dimensions for the input embeddings, and a single hidden layer with 750 dimensions. We use a vocabulary of 500 000 words (70 for the output vocabulary of P_l), from which we draw 100 noise samples for NCE (50 for P_l). We train for two epochs, each epoch being a full traversal of the training text. For unknown words, we back-off to a special *unk* token for the sequence models and P_l, and to the pre-terminal symbol for the other dependency models. We report perplexity values with softmax normalization, but disable normalization during decoding, relying on the self-normalization of NCE for efficiency. For the translation experiments with DLM and RDLM, we set the sibling window size q to 1, and the ancestor window size r to 2.[7]

We train baseline language models with interpolated modified Kneser-Ney smoothing with SRILM (Stolcke, 2002). The model in the SMT baseline uses the full vocabulary and a linear interpolation of component models for domain adaptation. For the perplexity evaluation, we use the same vocabulary and training data as for the Neural Network models.

For the English→German SMT evaluation, our baseline system is a string-to-tree SMT system with Moses (Koehn et al., 2007), with dependency parsing of the German texts (Sennrich et al., 2013). It is described in more detail in (Williams et al., 2014). This setup was ranked 1–2 (out of 18) in the WMT 2014 shared translation task and is state-of-the art. Our biggest deviation from this setup is that we do not enforce the morphological agreement constraints that are provided by a unification grammar (Williams and Koehn, 2011), but use them for analysis instead. For English→Russian, we copy the language-independent settings from the the English→German set-up, and perform dependency parsing with a Russian model for the Maltparser (Nivre et al., 2006; Sharoff and Nivre, 2011), applying projectivization after parsing.

We tune our system on a development set of 2000 sentences with k-best batch MIRA (Cherry and Foster, 2012) on BLEU and a linear interpolation of BLEU and HWCM$_f$, and report both scores for evaluation. We also report METEOR (Denkowski and Lavie, 2011) for German and TER (Snover et al., 2006). We control for optimizer instability by running the optimization three times per system and performing significance testing with Multeval (Clark et al., 2011), which we enhanced to also perform significance testing for HWCM$_f$.

5.2 Implementation notes on model by Shen et al. (2010)

We reimplement the model by Shen et al. (2010) for our evaluation. The authors did not specify training and smoothing of their model, so we only adopt their definition of the context window, and use the same neural network architecture as for our other models. Specifically, we use two neural networks: one for left dependents, and one for right dependents. We use maximum-likelihood estimation for the head of root nodes, ignoring unseen events. To distinguish between parents and siblings in the context window, we double the input vocabulary and mark parents with a suffix. Like Shen et al. (2010), we ignore the

[7] On our test set, a node has an average of 4.6 ancestors ($\sigma = 2.5$), and 1.2 left siblings ($\sigma = 1.3$).

language model	perplexity ref.	perplexity 1-best	entropy difference
5-gram (KN)	232.9	183.3	-4.4%
5-gram NNLM	207.3	207.5	0.0%
Shen et al. (2010)	345.1	383.0	1.8%
DLM (q=1; r=1)	213.7	259.9	3.6%
DLM (q=1; r=2)	136.9	188.3	6.5%
RDLM (q=1; r=2)	349.2	734.6	12.7%
RDLM, P_w	58.1	85.1	9.4%
RDLM, P_l	6.0	8.6	20.1%

Table 2: Perplexity of different Neural Network language models (and baseline with Kneser-Ney smoothing) on German reference translation (newstest2013) and baseline English→German translation output. Our goal is a language model that prefers the reference over the translation hypothesis, indicated by a lower perplexity and a positive entropy difference.

prediction of $STOP$ labels, meaning that our implementation assumes the dependency topology to be given. We use a trigram model like the original authors. Peter et al. (2012) experiment with higher orders variants, but do not consider grandparent nodes. We consider scalability to a larger ancestor context a real concern, since another duplication of the vocabulary may be necessary for each ancestor level.

5.3 Perplexity

There are a number of factors that make a direct comparison of the reference set perplexity unfair. Mainly, the unlabelled dependency model DLM and the one by Shen et al. (2010) assume that the dependency topology is given; P_w even assumes this for the dependency labels D. Conversely, the full RDLM predicts the terminal sequence, the dependency labels, and the dependency topology, and we thus expect it to have a higher perplexity.[8] Also note that we compare 5-gram n-gram models to 3- and 4-gram dependency models. A more minor difference is that n-gram models also predict end-of-sentence tokens, which the dependency models do not.

Rather than directly comparing perplexity between different models, our focus lies on a perplexity comparison between a human reference translation and the 1-best SMT output of a baseline translation system. Our basic assumption is that the difference in perplexity (or cross-entropy) tells us whether a model contains information that is not already part of the baseline model, and if incorporating it into our SMT system can nudge the system towards producing a translation that is more similar to the reference.

Results for English→German are shown in table 2. The baseline 5-gram language model with Kneser-Ney smoothing prefers the SMT output over the reference translation, which is natural given that this language model is part of the system producing the SMT output. The 5-gram NNLM improves over the Kneser-Ney models, and happens to assign almost the same perplexity score to both texts. This still means that it is less biased towards the SMT output than the baseline model, and can be a valuable addition to the model.

The dependency language models all show a preference for the reference translation, with DLM having a stronger preference than the model by Shen et al. (2010), and RDLM having the strongest preference. The direct comparison of DLM and P_w, which is the component of RDLM that predicts the terminal symbols, shows that dependency labels serve as a strong signal for predicting the terminals, confirming our initial hypothesis. The prediction of the dependency topology and labels through P_l means that the full RDLM has the highest perplexity of all models. However, it also strongly prefers the human reference text over the baseline SMT output.

5.4 Translation Quality

Translation results for English→German with different language models added to our baseline are shown in Table 3. Considering the systems tuned on BLEU, we observe that the 5-gram NNLM and RDLM are best in terms of BLEU and TER, but that RDLM is the only winner[9] according to HWCM$_f$ and METEOR. In particular, we observe a sizable gap of 0.6 HWCM$_f$ points between the NNLM and the RDLM systems, despite similar BLEU scores. The unlabelled DLM and the dependency LM by Shen et al. (2010), which are generally weaker than RDLM, also tend to improve HWCM$_f$ more than BLEU. This reflects the fact that the dependency

[8] For better comparability, we measure perplexity per surface word, not per prediction.

[9] We denote a system a winner if no other system [in the group of systems under consideration] is significantly better according to significance testing with Multeval.

MIRA objective	system	dev				newstest2013				newstest2014			
		BLEU	HWCM$_f$	METEOR	TER	BLEU	HWCM$_f$	METEOR	TER	BLEU	HWCM$_f$	METEOR	TER
BLEU	baseline	34.4	32.6	52.5	47.4	19.8	22.8	39.7*	62.4	20.3	23.2	42.0*	62.7
	5-gram NNLM	**35.3**	33.1	**53.2***	**46.4**	**20.4**	23.2	40.2	**61.7**	**21.0**	23.5	42.5*	**62.2**
	Shen et al. (2010)	34.4*	33.2	52.7*	46.9	20.0	23.2	40.0*	62.3	20.4	23.5	42.3*	62.9
	DLM	34.9*	**33.8**	**53.1***	46.8	20.3	23.6	40.1*	**61.7**	20.8	23.9	42.3*	**62.2**
	RDLM	35.0	**33.9**	**53.1***	46.7	**20.5**	**23.8**	**40.4***	**61.7**	**21.0**	**24.1**	**42.7***	**62.2**
	5-gram + RDLM	35.5	34.0	53.4*	46.3	20.7	23.7	40.6*	61.5	21.4	24.1	42.9*	61.7
BLEU + HWCM$_f$	baseline	34.4	33.0*	52.4	46.9*	20.0*	23.0*	39.6	61.9*	20.5*	23.3*	41.8	62.2*
	5-gram NNLM	**35.2**	33.5*	**53.0**	**46.0***	20.6*	23.4*	40.1	60.9*	21.1*	23.6	42.3	61.5*
	Shen et al. (2010)	34.2	33.8*	52.4	46.4*	20.2*	23.5*	39.8	61.8*	20.7*	23.7*	42.1	62.2*
	DLM	34.8	34.3*	52.7	**45.9***	20.4	23.8*	39.8	**60.7***	21.4*	24.2*	42.0	**60.9***
	RDLM	34.9	**34.5***	**53.0**	**45.8***	**20.9***	**24.2***	**40.3**	**60.7***	**21.6***	**24.5***	**42.5**	**60.8***
	5-gram + RDLM	35.4	34.6*	53.2	45.4*	21.0*	24.1*	40.4	60.5*	21.8*	24.4*	42.7	60.6*

Table 3: Translation quality of English→German string-to-tree SMT system with different language models, with k-best batch MIRA optimization on BLEU and BLEU+HWCM$_f$. Average of 3 optimization runs. **bold**: no other system in same block is significantly better ($p < 0.05$); *: significantly better than same model with other MIRA objective ($p < 0.05$). Higher scores are better for BLEU, HWCM$_f$ and METEOR; lower scores are better for TER.

LMs improve fluency along the syntactic n-grams that HWCM measures, whereas NNLM only improves local fluency, to which BLEU is most sensitive. The fact that the models cover different phenomena is also reflected in the fact that we see further gains from combining the 5-gram NNLM with the strongest dependency LM, RDLM, for a total improvement of 0.9–1.1 BLEU over the baseline.

If we use BLEU+HWCM$_f$ as our tuning objective, the difference between the models increases. Compared to the 5-gram NNLM, the RDLM system gains 0.8–0.9 points in HWCM$_f$ and 0.3–0.5 points in BLEU. Compared to the original baseline, tuned only on BLEU, the system with RDLM that is tuned on BLEU+HWCM$_f$ yields an improvement of 1.1–1.3 BLEU and 1.3–1.4 HWCM$_f$.

If we compare the same system being trained on both tuning objectives, we observe that tuning on BLEU+HWCM$_f$, unsurprisingly, yields higher HWCM$_f$ scores than tuning on BLEU only. What is more surprising is that adding HWCM$_f$ as a tuning objective also yields significantly higher BLEU on the test sets for 9 out of 10 data points. The gap is larger for the two systems with RDLM (0.3–0.6 BLEU) than for the baseline or the NNLM system (0.1–0.2 BLEU). We hypothesize that the inclusion of HWCM$_f$ as a tuning metric reduces overfitting and encourages the production of more grammatically well-formed constructions, which we expect to be a robust objective across different texts, especially when coupled with a strong dependency language model such as RDLM.

Some example translations are shown in table 4. They illustrate three error types in the baseline system:

1. an error in subject-verb agreement.

2. a subcategorisation error: *gelten* is a valid translation of the intransitive meaning of *apply*, but cannot be used for transitive constructions, where *anwenden* is correct.

3. a collocation error: two separate collocations are conflated in the baseline translation:

   - *reach a decision on [...]*
     eine Entscheidung über [...] treffen
   - *reach an agreement on [...]*
     eine Einigung über [...] erzielen

All errors are due to inter-dependencies in the sentence that have string-level gaps, but which can be modelled through syntactic n-grams, and are corrected by the system with RDLM and tuning on BLEU+HWCM$_f$.

We evaluate a subset of the systems on an English→Russian task to test whether the improvements from adding RDLM and tuning on BLEU+HWCM$_f$ apply to other language pairs. Results are shown in Table 5. The system with RDLM

1	source	also **the user** manages his identity and **can** therefore be anonymous.
	baseline	auch **der Benutzer** verwaltet seine Identität und **können** daher anonym sein.
	best	auch **der Benutzer** verwaltet seine Identität und **kann** daher anonym sein.
	reference	darüber hinaus verwaltet **der Inhaber** seine Identität und **kann** somit anonym bleiben.
2	source	how do you **apply this definition** to their daily life and social networks?
	baseline	wie kann man **diese Definition** für ihr tägliches Leben und soziale Netzwerke **gelten**?
	best	wie kann man **diese Definition** auf ihren Alltag und sozialen Netzwerken **anwenden**?
	reference	wie wird **diese Definition** auf seinen Alltag und die sozialen Netzwerke **angewendet**?
3	source	the City Council must **reach a decision** on this in December.
	baseline	Der Stadtrat muss im Dezember **eine Entscheidung** darüber **erzielen**.
	best	Im Dezember muss der Stadtrat **eine Entscheidung** darüber **treffen**.
	reference	Im Dezember muss dann noch die Stadtverordnetenversammlung **entscheiden**.

Table 4: SMT output of baseline system and best system (RDLM tuned on BLEU+HWCM$_f$).

MIRA objective	system	dev			newstest2013			newstest2014		
		BLEU	HWCM$_f$	TER	BLEU	HWCM$_f$	TER	BLEU	HWCM$_f$	TER
BLEU	baseline	22.5	21.6	56.7	17.1	18.8	64.7	25.9	23.9	54.5
	DLM	**23.3***	23.5	**56.0**	**17.5**	20.2	64.0	26.4	26.1	**53.8**
	RDLM	**23.1**	**23.7**	**56.0**	**17.6**	**20.4**	**63.8**	**26.6**	**26.5**	**53.7**
BLEU+ HWCM$_f$	baseline	22.5	22.9*	56.1*	17.2	19.7*	63.9*	25.8	25.1*	54.1*
	DLM	**23.0**	24.1*	**55.6***	**17.6**	**20.8***	**63.2***	26.4	26.9*	53.3*
	RDLM	**23.1**	**24.4***	**55.4***	**17.6**	**20.9***	**63.1***	**26.8***	**27.3***	**53.0***

Table 5: Translation quality of English→Russian string-to-tree SMT system with DLM and RDLM, with k-best batch MIRA optimization on BLEU and BLEU+HWCM$_f$. Average of 3 optimization runs. **bold**: no other system in same block is significantly better ($p < 0.05$); *: significantly better than same model with other MIRA objective ($p < 0.05$). Higher scores are better for BLEU and HWCM$_f$; lower scores are better for TER.

is the consistent winner, and significantly outperforms the baseline for all metrics and test sets. Tuning on BLEU+HWCM$_f$ results in further improvements in HWCM$_f$ and TER. Looking at the combined effect of adding RDLM and changing the tuning objective, we observe gains in BLEU by 0.5–0.9 points, and gains in HWCM$_f$ by 2.1–3.4 points.

5.5 Morphological Agreement

We argue that the dependency language models and HWCM$_f$ as a tuning metric improve grammaticality, and we are able to quantify one aspect thereof, morphological agreement, for English→German. Williams and Koehn (2011) introduce a unification grammar with hand-crafted agreement constraints to identify and suppress selected morphological agreement violations in German, namely in regards to noun phrase agreement, prepositional phrase agreement, and subject-verb agreement. We can use their grammar to analyse the effect of different models on morphological agreement by counting the number of translations that violate at least one agreement constraint. We assume that the number of false positives (i.e. correct analyses that trigger an agreement violation) remains roughly constant throughout all systems, so that a reduction in the number of agreement violations is an indicator of better grammatical agreement.

system	MIRA objective	
	BLEU	BLEU+HWCM$_f$
baseline	1028	1018
5-gram NNLM	845	825
Shen et al. (2010)	884	844
DLM	680	599
RDLM	550	468
5-gram + RDLM	576	484

Table 6: Number of English→German translation hypotheses with at least one agreement error according to unification grammar (Williams and Koehn, 2011) on newstest2013 (3000 sentences). Average of three MIRA runs.

Table 6 shows the results. While the 5-gram NNLM reduces the number of agreement errors somewhat compared to the baseline (-18%), the reduction is greater for DLM (-34%) and RDLM (-46%). Neither the baseline nor the 5-gram NNLM

profits strongly from tuning on HWCM$_f$, while the number of agreement errors is further reduced for the system with DLM (-41%) and RDLM (-54%). Adding the 5-gram NNLM to the RDLM system yields no further reduction on the number of agreement errors.

Enforcing the agreement constraints on the baseline system (tuned on BLEU+HWCM$_f$) provides us with a gain of 0.3 in both BLEU and HWCM$_f$; on the RDLM system, only 0.03. The fact that the benefit of enforcing the agreement constraints drops off more sharply than the number of constraint violations indicates that the remaining violations tend to be harder for the model to correct, e.g. because the translation model has not learned to produce the required inflection of a word, or because some of the remaining violations are false positives. While the dependency language models' effect of improving morphological agreement is not (fully) cumulative with the benefit from enforcing the unification constraints formulated by Williams and Koehn (2011), our model has the advantage of being language-independent, learning from the data itself rather than relying on manually developed, grammar-specific constraints, and covering a wider range of phenomena such as subcategorisation and syntactic collocations.

The results confirm that the RDLM is more effective at reducing morphological agreement errors than a similarly trained n-gram NNLM and the unlabelled DLM, and that adding HWCM$_f$ to the training objective is beneficial. On a a meta-evaluation level, we compare the rank correlation between the automatic metrics and the numer of agreement errors with Kendall's τ correlation, and observe that he number of agreement errors is more strongly (negatively) correlated with HWCM$_f$ ($\tau = -0.92$) than with BLEU ($\tau = -0.77$), METEOR ($\tau = -0.54$) or TER ($\tau = 0.69$). This supports our theoretical expectation that HWCM$_f$ is more sensitive to morphological agreement, which is enforced along syntactic n-grams, than n-gram metrics such as BLEU, or the unigram metric METEOR.

6 Related Work

While there has been a wide range of dependency language models proposed (e.g. (Chelba et al., 1997; Quirk et al., 2004; Shen et al., 2010; Zhang, 2009; Popel and Marecek, 2010)), there are vast differences in modelling assumptions. Our work is most similar to the dependency language model described in Shen et al. (2010), or the h-gram model proposed by Zhang (2009), both of which have been used for SMT. We make different modelling assumptions, relying less on configurational information, but including the prediction of dependency labels in the model. We argue that our relational modelling assumptions are more suitable for languages with a relatively free word order such as German.

To a lesser extent, our work is similar to other parsing models that have been used for language modelling, such as lexicalized PCFGs (Charniak, 2001; Collins, 2003; Charniak et al., 2003), or structured language models (Chelba and Jelinek, 2000); previous efforts to include them in the translation process failed to improve translation performance (Och et al., 2004; Post and Gildea, 2008). Differences in our work that could explain why we see improvements include the use of Neural Networks for training the model on the automatically parsed training text, instead of re-using existing parser models, which could be seen as a form of self-training (McClosky et al., 2006), and the integration of the language model into the decoder instead of n-best reranking. Also, there are major differences in the parsing models themselves. For instance, note that the structured LM by Chelba and Jelinek (2000) uses a binary branching structure, and that complex label sets would be required to encode subcategorisation frames in binary trees (Hockenmaier and Steedman, 2002).

Our neural network is a standard feed-forward neural network as introduced by Bengio et al. (2003). Recently, recursive neural networks have been proposed for syntactic parsing (Socher et al., 2010; Le and Zuidema, 2014). The recursive nature of such models allows for the encoding of more context; for an efficient integration into the dynamic programming search of SMT decoding, we deem our model, which makes stronger Markov assumptions, more suitable.

While BLEU has been the standard objective function for tuning the log-linear parameters in SMT systems, recent work has investigated alternative objective functions. Some authors concluded that none of

the tested alternatives could consistently outperform BLEU (Cer et al., 2010; Callison-Burch et al., 2011). Liu et al. (2011) report that tuning on the TESLA metric gives better results than tuning on BLEU; Lo et al. (2013) do the same for MEANT.

There is related work on improving morphological agreement and subcategorisation through post-editing (Rosa et al., 2012) or independent models for inflection generation (Toutanova et al., 2008; Weller et al., 2013). The latter models initially produce a stemmed translation, then predict the inflection through feature-rich sequence models. Such a pipeline of prediction steps is less powerful than our joint prediction of stems and inflection. For instance, in example 2 in Table 4, our model chooses a different stem to match the subcategorisation frame of the translation; it is not possible to fix the baseline translation with inflection changes alone.

7 Conclusion

The main contribution of this paper is the description of a relational dependency language model.[10] We show that it is a valuable asset to a state-of-the-art SMT system by comparing perplexity values with other types of languages models, and by its integration into decoding, which results in improvements according to automatic MT metrics and reduces the number of agreement errors. We show that the disfluencies that our model captures are qualitatively different from an n-gram Neural Network language model, with our model being more effective at modelling fluency phenomena along syntactic n-grams.

A second important contribution is the optimization of the log-linear parameters of an SMT system based on syntactic n-grams. We are to our knowledge the first to tune an SMT system on a non-shallow syntactic similarity metric. Apart from showing improvements by tuning on HWCM$_f$, our results also shed light on the interaction between models and tuning metrics. With n-gram language models, the choice of tuning metric only had a small effect on the English→German translation results. Only with dependency language models, which are able to model the syntactic n-grams that HWCM scores, did we see large improvements from adding HWCM$_f$ to the objective function. On the one hand, this has implications when evaluating new model components: using an objective function that cannot capture the impact of a model component can result in false negatives because the model component will not receive an appropriate weight, and the model may thus seem to be of little use, even in a human evaluation. On the other hand, it is an important finding for the evaluation of objective functions: the performance of an objective function is tied to the power of the underlying model. Without a model that is able to model syntactic n-grams, we might have concluded that HWCM is of little help as an objective function. Now, we hypothesize that HWCM is well-suited to optimize dependency language models because both operate on syntactic n-grams, just like BLEU and n-gram models are natural counterparts.

The approach we present is language-independent, and we evaluated it on SMT into German and Russian. While we have no empirical data on the model's effectiveness for other target languages, we suspect that syntactic n-grams are especially suited for modelling and evaluating translations into languages with inter-dependencies between distant words and relatively free word order, such as German, Czech, or Russian.

In this work, we relied on parse hypotheses being provided by a string-to-tree SMT decoder, but other settings are conceivable for future work, such as performing n-best string reranking by coupling the relational dependency LM with a monolingual parse algorithm. Another obvious extension of the relational dependency LM is the inclusion of more context, for instance through larger windows for siblings and ancestors, or source-context as in (Devlin et al., 2014). Also, we believe that the model can benefit from further advances in neural network modelling, for instance recent findings that ensembles of networks outperform a single network (Mikolov et al., 2011; Devlin et al., 2014)

[10] We have released an implementation of RDLM and tuning on HWCM$_f$ as part of the Moses decoder.

Acknowledgements

I thank Bonnie Webber and the anonymous reviewers for their helpful suggestions and feedback. This research was funded by the Swiss National Science Foundation under grant P2ZHP1_148717.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Daniel Cer, Christopher D. Manning, and Daniel Jurafsky. 2010. The Best Lexical Metric for Phrase-based Statistical MT System Optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 555–563, Los Angeles, California. Association for Computational Linguistics.

Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for statistical machine translation. In *MT Summit IX*, New Orleans, USA.

Eugene Charniak. 2001. Immediate-head Parsing for Language Models. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 124–131. Association for Computational Linguistics.

Ciprian Chelba and Frederick Jelinek. 2000. Structured language modeling. *Computer Speech & Language*, 14(4):283–332.

Ciprian Chelba, David Engle, Frederick Jelinek, Victor Jimenez, Sanjeev Khudanpur, Lidia Mangu, Harry Printz, Eric Ristad, Ronald Rosenfeld, Andreas Stolcke, and Dekai Wu. 1997. Structure And Performance Of A Dependency Language Model. In *Proceedings of Eurospeech*, pages 2775–2778.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 427–436, Montreal, Canada. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 176–181, Portland, Oregon. Association for Computational Linguistics.

Michael Collins. 2003. Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29:589 – 637.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland. Association for Computational Linguistics.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2002. Generative Models for Statistical Parsing with Combinatory Categorial Grammar. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 335–342, Philadelphia, PA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL-2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Phong Le and Willem Zuidema. 2014. The Inside-Outside Recursive Neural Network model for Dependency Parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 729–739, Doha, Qatar. Association for Computational Linguistics.

Ding Liu and Daniel Gildea. 2005. Syntactic Features for Evaluation of Machine Translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic*

Evaluation Measures for Machine Translation and/or Summarization, pages 25–32, Ann Arbor, Michigan.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2011. Better Evaluation Metrics Lead to Better Machine Translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 375–384, Edinburgh, UK.

Chi-kiu Lo, Meriem Beloucif, and Dekai Wu. 2013. Improving machine translation into Chinese by tuning against Chinese MEANT. In *10th International Workshop on Spoken Language Translation (IWSLT 2013)*, Heidelberg, Germany.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-training for Parsing. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 152–159, New York. Association for Computational Linguistics.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP 2011, pages 5528–5531, Prague, Czech Republic.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 2216 –2219, Genoa, Italy.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 161–168, Boston, Massachusetts, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, PA. Association for Computational Linguistics.

Jan-Thorsten Peter, Matthias Huck, Hermann Ney, and Daniel Stein. 2012. Soft String-to-Dependency Hierarchical Machine Translation. In *Informatiktage der Gesellschaft für Informatik*, Lecture Notes in Informatics (LNI), pages 59–62, Bonn, Germany. Gesellschaft für Informatik.

Martin Popel and David Marecek. 2010. Perplexity of n-Gram and Dependency Language Models. In Petr Sojka, Ales Horák, Ivan Kopecek, and Karel Pala, editors, *TSD*, volume 6231 of *Lecture Notes in Computer Science*, pages 173–180. Springer.

Matt Post and Daniel Gildea. 2008. Parsers as language models for statistical machine translation. In *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2004. Dependency Tree Translation: Syntactically Informed Phrasal SMT. Technical Report MSR-TR-2004-113, Microsoft Research.

Rudolf Rosa, David Mareček, and Ondřej Dušek. 2012. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 362–368, Montreal, Canada. Association for Computational Linguistics.

Rico Sennrich, Martin Volk, and Gerold Schneider. 2013. Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2013*, pages 601–609, Hissar, Bulgaria.

Rico Sennrich. 2014. A CYK+ Variant for SCFG Decoding Without a Dot Chart. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 94–102, Doha, Qatar, October. Association for Computational Linguistics.

Serge Sharoff and Joakim Nivre. 2011. The proper place of men and machines in language technology. Processing Russian without any linguistic knowledge. In *Proceedings of the International Conference on Computational Linguistics and Artificial Intelligence Dialog 2011*.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2010. String-to-dependency Statistical Machine Translation. *Comput. Linguist.*, 36(4):649–671.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2013. Syntactic Dependency-based N-grams As Classification Features. In *Proceedings of the 11th Mexican International Conference on Advances in Computational Intelligence - Volume Part II*, MICAI'12, pages 1–11, Berlin, Heidelberg. Springer-Verlag.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2010. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks. *Proceedings of the Deep Learning and Unsupervised Feature Learning Workshop of NIPS 2010*, pages 1–9.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado, USA.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying Morphology Generation Models to Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Reut Tsarfaty, Khalil Sima'an, and Remko Scha. 2009. An Alternative to Head-Driven Approaches for Parsing a (Relatively) Free Word-Order Language. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 842–851, Singapore.

Reut Tsarfaty. 2010. *Relational-Realizational Parsing*. Ph.D. thesis, University of Amsterdam.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with Large-Scale Neural Language Models Improves Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 1387–1392, Seattle, Washington, USA.

Marion Weller, Alexander Fraser, and Sabine Schulte im Walde. 2013. Using subcategorization knowledge to improve case prediction for translation to German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 593–603, Sofia, Bulgaria. Association for Computational Linguistics.

Philip Williams and Philipp Koehn. 2011. Agreement Constraints for Statistical Machine Translation into German. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 217–226, Edinburgh, UK. Association for Computational Linguistics.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Eva Hasler, and Philipp Koehn. 2014. Edinburgh's Syntax-Based Systems at WMT 2014. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 207–214, Baltimore, Maryland, USA. Association for Computational Linguistics.

Joy Ying Zhang. 2009. *Structured Language Models for Statistical Machine Translation*. Ph.D. thesis, Johns Hopkins University.