Monolingual Distributional Similarity for Text-to-Text Generation

Juri Ganitkevitch, Benjamin Van Durme, and **Chris Callison-Burch**
Center for Language and Speech Processing
Human Language Technology Center of Excellence
Johns Hopkins University
Baltimore, MD 21218, USA

Abstract

Previous work on paraphrase extraction and application has relied on either parallel datasets, or on distributional similarity metrics over large text corpora. Our approach combines these two orthogonal sources of information and directly integrates them into our paraphrasing system's log-linear model. We compare different distributional similarity feature-sets and show significant improvements in grammaticality and meaning retention on the example text-to-text generation task of sentence compression, achieving state-of-the-art quality.

1 Introduction

A wide variety of applications in natural language processing can be cast in terms of text-to-text generation. Given input in the form of natural language, a text-to-text generation system produces natural language output that is subject to a set of constraints. Compression systems, for instance, produce shorter sentences. Paraphrases, i.e. differing textual realizations of the same meaning, are a crucial components of text-to-text generation systems, and have been successfully applied to tasks such as multi-document summarization (Barzilay et al., 1999; Barzilay, 2003), query expansion (Anick and Tipirneni, 1999; Riezler et al., 2007), question answering (McKeown, 1979; Ravichandran and Hovy, 2002), sentence compression (Cohn and Lapata, 2008; Zhao et al., 2009), and simplification (Wubben et al., 2012).

Paraphrase collections for text-to-text generation have been extracted from a variety of different corpora. Several approaches rely on bilingual parallel data (Bannard and Callison-Burch, 2005; Zhao et al., 2008; Callison-Burch, 2008; Ganitkevitch et al., 2011), while others leverage distributional methods on monolingual text corpora (Lin and Pantel, 2001; Bhagat and Ravichandran, 2008). So far, however, only preliminary studies have been undertaken to combine the information from these two sources (Chan et al., 2011).

In this paper, we describe an extension of Ganitkevitch et al. (2011)'s bilingual data-based approach. We augment the bilingually-sourced paraphrases using features based on monolingual distributional similarity. More specifically:

- We show that using monolingual distributional similarity features improves paraphrase quality beyond what we can achieve with features estimated from bilingual data.
- We define distributional similarity for paraphrase patterns that contain constituent-level gaps, e.g.

  $sim(\text{one } JJ \text{ instance of } NP, \text{a } JJ \text{ case of } NP)$.

  This generalizes over distributional similarity for contiguous phrases.
- We compare different types of monolingual distributional information and show that they can be used to achieve significant improvements in grammaticality.
- Finally, we compare our method to several strong baselines on the text-to-text generation task of sentence compression. Our method shows state-of-the-art results, beating a purely bilingually sourced paraphrasing system.

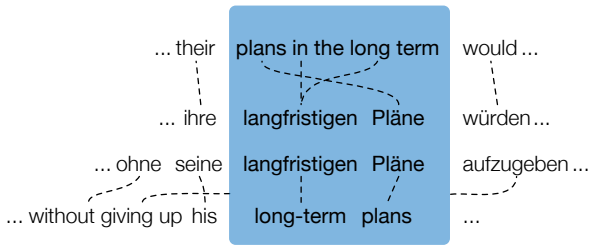


Figure 1: Pivot-based paraphrase extraction for contiguous phrases. Two phrases translating to the same phrase in the foreign language are assumed to be paraphrases of one another.

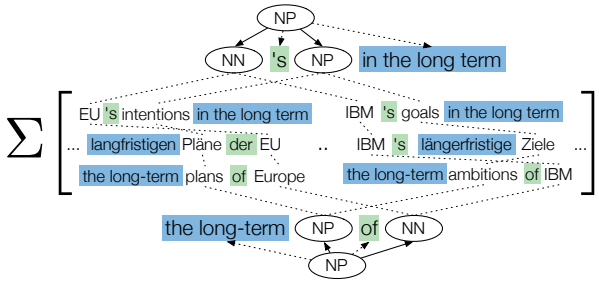


Figure 2: Extraction of syntactic paraphrases via the pivoting approach: We aggregate over different surface realizations, matching the lexicalized portions of the rule and generalizing over the nonterminals.

2 Background

Approaches to paraphrase extraction differ based on their underlying data source. In Section 2.1 we outline pivot-based paraphrase extraction from bilingual data, while the contextual features used to determine closeness in meaning in monolingual approaches is described in Section 2.2.

2.1 Paraphrase Extraction via Pivoting

Following Ganitkevitch et al. (2011), we formulate our paraphrases as a syntactically annotated *synchronous context-free grammar* (SCFG) (Aho and Ullman, 1972; Chiang, 2005). An SCFG rule has the form:

$$\mathbf{r} = C \rightarrow \langle f, e, \sim, \vec{\varphi} \rangle,$$

where the left-hand side of the rule, C, is a nonterminal and the right-hand sides f and e are strings of terminal and nonterminal symbols. There is a one-to-one correspondency between the nonterminals in f and e: each nonterminal symbol in f has to also appear in e. The function $\sim$ captures this bijective mapping between the nonterminals. Drawing on machine translation terminology, we refer to f as the *source* and e as the *target* side of the rule.

Each rule is annotated with a feature vector of feature functions $\vec{\varphi} = \{\varphi_1 ... \varphi_N\}$ that, using a corresponding weight vector $\vec{\lambda}$, are combined in a log-linear model to compute the *cost* of applying $\mathbf{r}$:

$$cost(\mathbf{r}) = -\sum_{i=1}^{N} \lambda_i \log \varphi_i. \quad (1)$$

A wide variety of feature functions can be formulated. We detail the feature-set used in our experiments in Section 4.

To extract paraphrases we follow the intuition that two English strings e_1 and e_2 that translate to the same foreign string f can be assumed to have the same meaning, as illustrated in Figure 1.[1]

First, we use standard machine translation methods to extract a foreign-to-English translation grammar from a bilingual parallel corpus (Koehn, 2010). Then, for each pair of translation rules where the left-hand side C and foreign string f match:

$$\mathbf{r}_1 = C \rightarrow \langle f, e_1, \sim_1, \vec{\varphi}_1 \rangle$$
$$\mathbf{r}_2 = C \rightarrow \langle f, e_2, \sim_2, \vec{\varphi}_2 \rangle,$$

we *pivot* over f to create a paraphrase rule $\mathbf{r}_p$:

$$\mathbf{r}_p = C \rightarrow \langle e_1, e_2, \sim_p, \vec{\varphi}_p \rangle,$$

with a combined nonterminal correspondency function $\sim_p$. Note that the common source side f implies that e_1 and e_2 share the same set of nonterminal symbols.

The paraphrase feature vector $\vec{\varphi}_p$ is computed from the translation feature vectors $\vec{\varphi}_1$ and $\vec{\varphi}_2$ by following the pivoting idea. For instance, we estimate the conditional paraphrase probability $p(e_2|e_1)$ by marginalizing over all shared foreign-language translations f:

$$p(e_2|e_1) = \sum_f p(e_2, f|e_1) \quad (2)$$
$$= \sum_f p(e_2|f, e_1) p(f|e_1) \quad (3)$$
$$\approx \sum_f p(e_2|f) p(f|e_1). \quad (4)$$

[1] See Yao et al. (2012) for an analysis of this assumption.

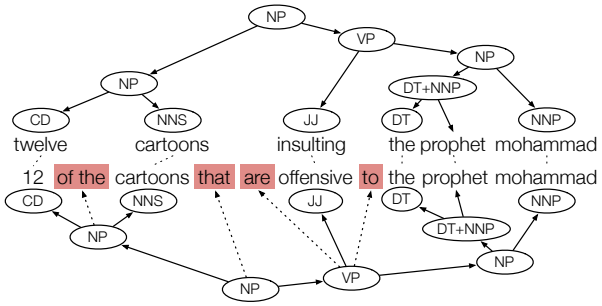


Figure 3: An example of a synchronous paraphrastic derivation, here a sentence compression. Shaded words are deleted in the indicated rule applications.

Figure 2 illustrates syntax-constrained pivoting and feature aggregation over multiple foreign language translations for a paraphrase pattern.

After the SCFG has been extracted, it can be used within standard machine translation machinery, such as the Joshua decoder (Ganitkevitch et al., 2012). Figure 3 shows an example for a synchronous paraphrastic derivation produced as a result of applying our paraphrase grammar in the decoding process.

The approach outlined relies on aligned bilingual texts to identify phrases and patterns that are equivalent in meaning. When extracting paraphrases from monolingual text, we have to rely on an entirely different set of semantic cues and features.

2.2 Monolingual Distributional Similarity

Methods based on monolingual text corpora measure the similarity of phrases based on contextual features. To describe a phrase e, we define a set of features that capture the context of an occurrence of e in our corpus. Writing the context vector for the i-th occurrence of e as $\vec{s}_{e,i}$, we can aggregate over all occurrences of e, resulting in a *distributional* signature for e, $\vec{s}_e = \sum_i \vec{s}_{e,i}$. Following the intuition that phrases with similar meanings occur in similar contexts, we can then quantify the goodness of e' as a paraphrase of e by computing the cosine similarity between their distributional signatures:

$$sim(e, e') = \frac{\vec{s}_e \cdot \vec{s}_{e'}}{|\vec{s}_e||\vec{s}_{e'}|}.$$

A wide variety of features have been used to describe the distributional context of a phrase. Rich, linguistically informed feature-sets that rely on dependency and constituency parses, part-of-speech tags, or lemmatization have been proposed in widely known work such as by Church and Hanks (1991) and Lin and Pantel (2001). For instance, a phrase is described by the various syntactic relations it has with lexical items in its context, such as: "for what verbs do we see with the phrase as the subject?", or "what adjectives modify the phrase?".

However, when moving to vast text collections or collapsed representations of large text corpora, linguistic annotations can become impractically expensive to produce. A straightforward and widely used solution is to fall back onto lexical n-gram features, e.g. "what words or bigrams have we seen to the left of this phrase?" A substantial body of work has focussed on using this type of feature-set for a variety of purposes in NLP (Lapata and Keller, 2005; Bhagat and Ravichandran, 2008; Lin et al., 2010; Van Durme and Lall, 2010).

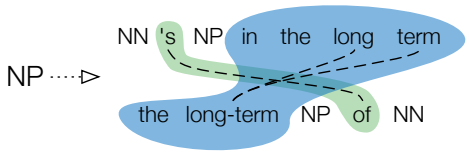
2.3 Other Related Work

Recently, Chan et al. (2011) presented an initial investigation into combining phrasal paraphrases obtained through bilingual pivoting with monolingual distributional information. Their work investigated a reranking approach and evaluated their method via a substitution task, showing that the two sources of information are complementary and can yield improvements in paraphrase quality when combined.

3 Incorporating Distributional Similarity

In order to incorporate distributional similarity information into the paraphrasing system, we need to calculate similarity scores for the paraphrastic SCFG rules in our grammar. For rules with purely lexical right-hand sides e_1 and e_2 this is a simple task, and the similarity score $sim(e_1, e_2)$ can be directly included in the rule's feature vector $\vec{\varphi}$. However, if e_1 and e_2 are long, their occurrences become sparse and their similarity can no longer be reliably estimated. In our case, the right-hand sides of our rules often contain gaps and computing a similarity score is less straightforward.

Figure 4 shows an example of such a discontinuous rule and illustrates our solution: we decompose the discontinuous patterns that make up the



$$sim(\mathbf{r}) = \frac{1}{2}\left(sim\begin{pmatrix}\text{the long-term}\\ \text{in the long term}\end{pmatrix} + sim\begin{pmatrix}\text{'s}\\ \text{of}\end{pmatrix}\right)$$

Figure 4: Scoring a rule by extracting and scoring contiguous phrases consistent with the alignment. The overall score of the rule is determined by averaging across all pairs of contiguous subphrases.

right-hand sides of a rule $\mathbf{r}$ into pairs of contiguous phrases $\mathcal{P}(\mathbf{r}) = \{\langle e, e'\rangle\}$, for which we can look up distributional signatures and compute similarity scores. This decomposition into phrases is non-trivial, since our sentential paraphrase rules often involve significant reordering or structural changes. To avoid comparing unrelated phrase pairs, we require $\mathcal{P}(\mathbf{r})$ to be consistent with a token alignment $\mathbf{a}$. The alignment is defined analogously to word alignments in machine translation, and computed by treating the source and target sides of our paraphrase rules as a parallel corpus.

We define the overall similarity score of the rule to be the average of the similarity scores of all extracted phrase pairs:

$$sim(\mathbf{r}, \mathbf{a}) = \frac{1}{|\mathcal{P}(\mathbf{a})|} \sum_{(e,e') \in \mathcal{P}(\mathbf{a})} sim(e, e').$$

Since the distributional signatures for long, rare phrases may be computed from only a handful of occurrences, we additionally query for the shorter sub-phrases that are more likely to have been observed often enough to have reliable signatures and thus similarity estimates.

Our definition of the similarity of two discontinuous phrases substantially differs from others in the literature. This difference is due to a difference in motivation. Lin and Pantel (2001), for instance, seek to find new paraphrase pairs by comparing their arguments. In this work, however, we try to add orthogonal information to existing paraphrase pairs. Both our definition of pattern similarity and our feature-set (see Section 4.3) are therefore geared towards comparing the substitutability and context similarity of a pair of paraphrases.

Our two similarity scores are incorporated into the paraphraser as additional rule features in $\vec{\varphi}$, sim_{ngram} and sim_{syn}, respectively. We estimate the corresponding weights along with the other λ_i as detailed in Section 4.

4 Experimental Setup

4.1 Task: Sentence Compression

To evaluate our method on a real text-to-text application, we use the sentence compression task. To tune the parameters of our paraphrase system for sentence compression, we need an appropriate corpus of reference compressions. Since our model is designed to compress by paraphrasing rather than deletion, the commonly used deletion-based compression data sets like the Ziff-Davis corpus are not suitable. We thus use the dataset introduced in our previous work (Ganitkevitch et al., 2011).

Beginning with 9570 tuples of parallel English–English sentences obtained from multiple reference translations for machine translation evaluation, we construct a parallel compression corpus by selecting the longest reference in each tuple as the source sentence and the shortest reference as the target sentence. We further retain only those sentence pairs where the compression ratio cr falls in the range $0.5 < cr \leq 0.8$. From these, we select 936 sentences for the development set, as well as 560 sentences for a test set that we use to gauge the performance of our system.

We contrast our distributional similarity-informed paraphrase system with a pivoting-only baseline, as well as an implementation of Clarke and Lapata (2008)'s state-of-the-art compression model which uses a series of constraints in an integer linear programming (ILP) solver.

4.2 Baseline Paraphrase Grammar

We extract our paraphrase grammar from the French–English portion of the Europarl corpus (version 5) (Koehn, 2005). The Berkeley aligner (Liang et al., 2006) and the Berkeley parser (Petrov and Klein, 2007) are used to align the bitext and parse the English side, respectively. The paraphrase grammar is produced using the Hadoop-based Thrax

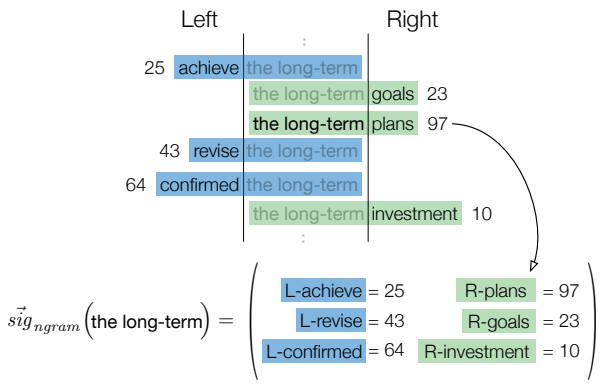


Figure 5: An example of the n-gram feature extraction on an n-gram corpus. Here, "the long-term" is seen preceded by "revise" (43 times) and followed by "plans" (97 times). The corresponding left- and right-side features are added to the phrase signature with the counts of the n-grams that gave rise to them.

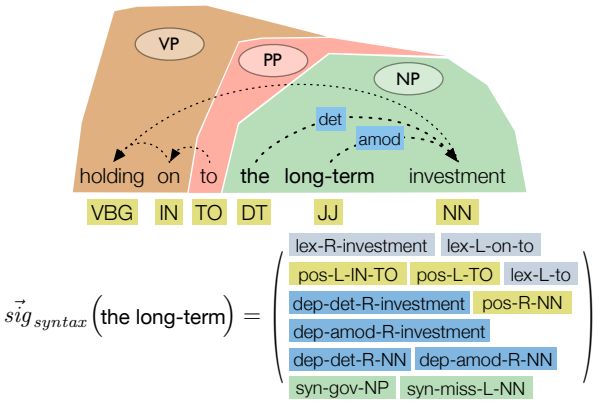


Figure 6: An example of the syntactic feature-set. The phrase "the long-term" is annotated with position-aware lexical and part-of-speech n-gram features (e.g. "on to" on the left, and "investment" and "NN" to its right), labeled dependency links (e.g. $amod - investment$) and features derived from the phrase's CCG label NP/NN.

grammar extractor's paraphrase mode (Ganitkevitch et al., 2012). The syntactic nonterminal labels we allowed in the grammar were limited to constituent labels and CCG-style slashed categories. Paraphrase grammars extracted via pivoting tend to grow very large. To keep the grammar size manageable, we pruned away all paraphrase rules whose phrasal paraphrase probabilities $p(e_1|e_2)$ or $p(e_2|e_1)$ were smaller than 0.001.

We extend the feature-set used in Ganitkevitch et al. (2011) with a number of features that aim to better describe a rule's compressive power: on top of the word count features $wcount_{src}$ and $wcount_{tgt}$ and the word count difference feature $wcount_{diff}$, we add character based count and difference features $ccount_{src}$, $ccount_{tgt}$, and $ccount_{diff}$, as well as log-compression ratio features $word_{cr} = \log \frac{wcount_{tgt}}{wcount_{src}}$ and the analogously defined $char_{cr} = \log \frac{ccount_{tgt}}{ccount_{src}}$.

For model tuning and decoding we used the Joshua machine translation system (Ganitkevitch et al., 2012). The model weights are estimated using an implementation of the PRO tuning algorithm (Hopkins and May, 2011), with PRÉCIS as our objective function (Ganitkevitch et al., 2011). The language model used in our paraphraser and the Clarke and Lapata (2008) baseline system is a Kneser-Ney discounted 5-gram model estimated on the Gigaword corpus using the SRILM toolkit (Stolcke, 2002).

4.3 Distributional Similarity Model

To investigate the impact of the feature-set used to construct distributional signatures, we contrast two approaches: a high-coverage collection of distributional signatures with a relatively simple feature-set, and a much smaller set of signatures with a rich, syntactically informed feature-set.

4.3.1 n-gram Model

The high-coverage model (from here on: n-gram model) is drawn from a web-scale n-gram corpus (Brants and Franz, 2006; Lin et al., 2010). We extract signatures for phrases up to a length of 4. For each phrase p we look at n-grams of the form wp and pv, where w and v are single words. We then extract the corresponding features w_{left} and v_{right}. The feature count is set to the count of the n-gram, reflecting the frequency with which p was preceded or followed, respectively, by w and v in the data the n-gram corpus is based on. Figure 5 illustrates this feature extraction approach. The resulting collection comprises distributional signatures for the 200 million most frequent 1-to-4-grams in the n-gram corpus.

4.3.2 Syntactic Model

For the syntactically informed signature model (from here on: syntax model), we use the constituency and dependency parses provided in the Annotated Gigaword corpus (Napoles et al., 2012). We limit ourselves to the Los Angeles Times/Washington Post portion of the corpus and extract phrases up to a length of 4. The following feature set is used to compute distributional signatures for the extracted phrases:

- Position-aware lexical and part-of-speech unigram and bigram features, drawn from a three-word window to the right and left of the phrase.
- Features based on dependencies for both links into and out of the phrase, labeled with the corresponding lexical item and POS. If the phrase corresponds to a complete subtree in the constituency parse we additionally include lexical and POS features for its head word.
- Syntactic features for any constituents governing the phrase, as well as for CCG-style slashed constituent labels for the phrase. The latter are split in governing constituent and missing constituent (with directionality).

Figure 6 illustrates the syntax model's feature extraction for an example phrase occurrence. Using this method we extract distributional signatures for over 12 million 1-to-4-gram phrases.

4.3.3 Locality Sensitive Hashing

Collecting distributional signatures for a large number of phrases quickly leads to unmanageably large datasets. Storing the syntax model's 12 million signatures in a compressed readable format, for instance, requires over 20GB of disk space. Like Ravichandran et al. (2005) and Bhagat and Ravichandran (2008), we rely on locality sensitive hashing (LSH) to make the use of these large collections practical.

In order to avoid explicitly computing the feature vectors, which can be memory intensive for frequent phrases, we chose the online LSH variant described by Van Durme and Lall (2010), as implemented in the Jerboa toolkit (Van Durme, 2012). This method, based on the earlier work of Indyk and Motwani (1998) and Charikar (2002), approximates the cosine similarity between two feature vectors based on the Hamming distance in a dimensionality-reduced bitwise representation. Two feature vectors u, v each of dimension d are first projected through a $d \times b$ random matrix populated with draws from $\mathcal{N}(0, 1)$. We then convert the resulting b-dimensional vectors into bit-vectors by setting each bit of the signature conditioned on whether the corresponding projected value is less than 0. Now, given the bit signatures $h(\vec{u})$ and $h(\vec{v})$, we can approximate the cosine similarity of u and v as:

$$sim'(u, v) = \cos\Big(\frac{D(h(\vec{u}), h(\vec{v}))}{b}\pi\Big),$$

where $d(\cdot, \cdot)$ is the Hamming distance. In our experiments we use 256-bit signatures. This reduces the memory requirements for the syntax model to around 600MB.

5 Evaluation Results

To rate the quality of our output, we solicit human judgments of the compressions along two five-point scales: grammaticality and meaning preservation. Judges are instructed to decide how much the meaning from a reference translation is retained in the compressed sentence, with a score of 5 indicating that all of the important information is present, and 1 being that the compression does not retain any of the original meaning. Similarly, a grammar score of 5 indicates perfect grammaticality, while a score of 1 is assigned to sentences that are entirely ungrammatical. We ran our evaluation on Mechanical Turk, where a total of 126 judges provided 3 redundant judgments for each system output. To provide additional quality control, our HITs were augmented with both positive and negative control compressions. For the positive control we used the reference compressions from our test set. Negative control was provided by adding a compression model based on random word deletions to the mix.

In Table 1 we compare our distributional similarity-augmented systems to the plain pivoting-based baseline and the ILP approach. The compression ratios of the paraphrasing systems are tuned to match the average compression ratio seen on the development and test set. The ILP system is config-

ured to loosely match this ratio, as to not overly constrain its search space. Our results indicate that the paraphrase approach significantly outperforms ILP on meaning retention. However, the baseline system shows notable weaknesses in grammaticality. Adding the n-gram distributional similarity model to the paraphraser recovers some of the difference in grammaticality while simultaneously yielding some gain in the compressions' meaning retention. Moving to distributional similarity estimated on the syntactic feature-set yields additional improvement, despite the model's lower coverage.

It is known that human evaluation scores correlate linearly with the compression ratio produced by a sentence compression system (Napoles et al., 2011). Thus, to ensure fairness in our comparisons, we produce a pairwise comparison breakdown that only takes into account compressions of almost identical length.[2] Figure 7 shows the results of this analysis, detailing the number of wins and ties in the human judgements.

We note that the gains in meaning retention over both the baseline and the ILP system are still present in the pairwise breakdown. The gains over the paraphrasing baseline, as well as the improvement in meaning over ILP are statistically significant at $p < 0.05$ (using the sign test).

We can observe that there is substantial overlap between the baseline paraphraser and the n-gram model, while the syntax model appears to yield noticeably different output far more often.

Table 2 shows two example sentences drawn from our test set and the compressions produced by the different systems. It can be seen that both the paraphrase-based and ILP systems produce good quality results, with the paraphrase system retaining the meaning of the source sentence more accurately.

6 Conclusion

We presented a method to incorporate monolingual distributional similarity into linguistically informed paraphrases extracted from bilingual parallel data. Having extended the notion of similarity to discontiguous pattern with multi-word gaps, we investigated the effect of using feature-sets of varying complexity to compute distributional similarity for our paraphrase collection. We conclude that, compared to a simple large-scale model, a rich, syntax-based feature-set, even with significantly lower coverage, noticeably improves output quality in a text-to-text generation task. Our syntactic method significantly improves grammaticality and meaning retention over a strong paraphrastic baseline, and offers substantial gains in meaning retention over a deletion-based state-of-the-art system.

Acknowledgements This research was supported in part by the NSF under grant IIS-0713448 and in part by the EuroMatrixPlus project funded by the European Commission (7th Framework Programme). Opinions, interpretations, and conclusions are the authors' alone.

[2] We require the compressions to be within $\pm 10\%$ length of one another.

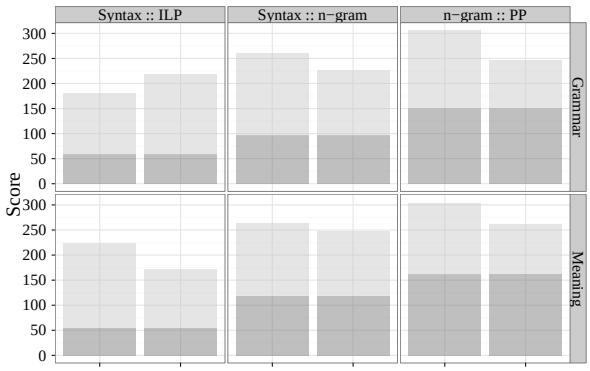


Figure 7: A pairwise breakdown of the human judgments comparing the systems. Dark grey regions show the number of times the two systems were tied, and light grey shows how many times one system was judged to be better than the other.

	CR	Meaning	Grammar
Reference	0.80	4.80	4.54
ILP	0.74	3.44	**3.41**
PP	0.78	3.53	2.98
PP + n-gram	0.80	3.65	3.16
PP + syntax	0.79	**3.70**	3.26
Random Deletions	0.78	2.91	2.53

Table 1: Results of the human evaluation on longer compressions: pairwise compression rates (CR), meaning and grammaticality scores. Bold indicates a statistically significance difference at $p < 0.05$.

Source	should these political developments have an impact on sports ?
Reference	should these political events affect sports ?
Syntax	should these events have an impact on sports ?
n-gram	these political developments impact on sports ?
PP	should these events impact on sports ?
ILP	political developments have an impact
Source	now we have to think and make a decision about our direction and choose only one way . thanks .
Reference	we should ponder it and decide our path and follow it , thanks .
Syntax	now we think and decide on our way and choose one way . thanks .
n-gram	now we have and decide on our way and choose one way . thanks .
PP	now we have and decide on our way and choose one way . thanks .
ILP	we have to think and make a decision and choose way thanks

Table 2: Example compressions produced by our systems and the baselines Table 1 for three input sentences from our test data.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation, and Compiling*. Prentice Hall.

Peter G. Anick and Suresh Tipirneni. 1999. The paraphrase search assistant: terminological feedback for iterative information seeking. In *Proceedings of SIGIR*.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of ACL*.

Regina Barzilay. 2003. *Information Fusion for Mutlidocument Summarization: Paraphrasing and Generation*. Ph.D. thesis, Columbia University, New York.

Rahul Bhagat and Deepak Ravichandran. 2008. Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL/HLT*.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP*.

Tsz Ping Chan, Chris Callison-Burch, and Benjamin Van Durme. 2011. Reranking bilingually extracted paraphrases using monolingual distributional similarity. In *EMNLP Workshop on GEMS*.

Moses Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of STOC*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*.

Kenneth Church and Patrick Hanks. 1991. Word association norms, mutual information and lexicography. *Computational Linguistics*, 6(1):22–29.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:273–381.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the COLING*.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In *Proceedings of EMNLP*.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post,

and Chris Callison-Burch. 2012. Joshua 4.0: Packing, PRO, and paraphrases. In *Proceedings of WMT12*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of EMNLP*.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of STOC*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Mirella Lapata and Frank Keller. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing*, 2(1).

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of HLT/NAACL*.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules from text. *Natural Language Engineering*.

Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. 2010. New tools for web-scale n-grams. In *Proceedings of LREC*.

Kathleen R. McKeown. 1979. Paraphrasing using given and new information in a question-answer system. In *Proceedings of ACL*.

Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme. 2011. Paraphrastic sentence compression with a character-based metric: Tightening without deletion. *Workshop on Monolingual Text-To-Text Generation*.

Courtney Napoles, Matt Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of AKBC-WEKEX 2012*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT/NAACL*.

Deepak Ravichandran and Eduard Hovy. 2002. Learning sufrace text patterns for a question answering system. In *Proceedings of ACL*.

Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. 2005. Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering. In *Proceedings of ACL*.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of ACL*.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceeding of the International Conference on Spoken Language Processing*.

Benjamin Van Durme and Ashwin Lall. 2010. Online generation of locality sensitive hash signatures. In *Proceedings of ACL, Short Papers*.

Benjamin Van Durme. 2012. Jerboa: A toolkit for randomized and streaming algorithms. Technical Report 7, Human Language Technology Center of Excellence, Johns Hopkins University.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of ACL*.

Xuchen Yao, Benjamin Van Durme, and Chris Callison-Burch. 2012. Expectations of word sense in parallel corpora. In *Proceedings of HLT/NAACL*.

Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. 2008. Pivot approach for extracting paraphrase patterns from bilingual corpora. In *Proceedings of ACL/HLT*.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of ACL*.