Monolingual Distributional Similarity for Text-to-Text Generation

Abstract

Previous work obtained collections of paraphrases by either relying on bilingual parallel datasets, or by using distributional similarity metrics over large text corpora. Our approach combines these two orthogonal sources of information and directly integrates them into the our paraphrasing system's log-linear model. We compare different distributional similarity feature sets and show significant improvements in output quality on the example text-to-text generation task of sentence compression, beating two state-of-the-art baselines.

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 fulfills previously defined constraints and objectives on both the text's surface form and meaning. 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 and simplification.

One way of using paraphrases for text-to-text generation is to appropriate the machinery developed for statistical machine translation (Quirk et al., 2004). Syntactically informed machine translation approaches have been used to create sentential paraphrasing systems. Both Cohn et al. (2008) and Ganitkevitch et al. (2011) described large-scale extraction methods for syntactically annotated paraphrases from bilingual parallel corpora. This framework can be adapted to many sentential text-to-text generation tasks.

In this paper, we describe an extension of Ganitkevitch et al. (2011)'s approach by introducing a new component into the paraphrasing system that is *not* derived from the statistical machine translation machinery. We use an orthogonal source of information: 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.
- Different types of monolingual distributional information can be used to achieve improvements in grammaticality or word sense disambiguation.
- We define distributional similarity for paraphrase patterns that contain gaps, e.g.

  $sim(NN \text{ ate } NP, NP \text{ was eaten by } NN)$.

  This generalizes over previous approaches that defined the notion for contiguous phrases or single-word gaps.
- Finally, we compare our method to several strong baselines on the text-to-text generation task of sentence compression. Our method beats a purely bilingually sourced paraphrasing system and an ILP-based compression model.

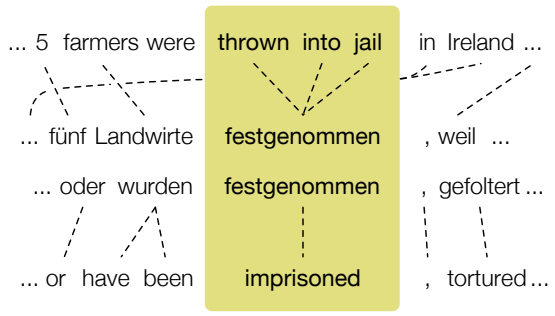

Figure 1: Pivot-based paraphrase extraction for contiguous phrases. Two phrases translating to the same phrase in the foreign language are assumed to be paraphrases of one another.

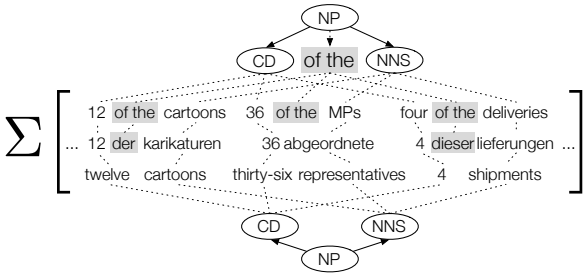

Figure 2: An example of syntactic paraphrase extraction via the pivoting approach. We sum over the multiple surface forms that give rise to the syntactic pattern.

2 Background

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 with an equal number of nonterminals. The function $\sim$ defines a one-to-one correspondency function between the nonterminals in f and e. 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 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 $\mathbf{r}$:

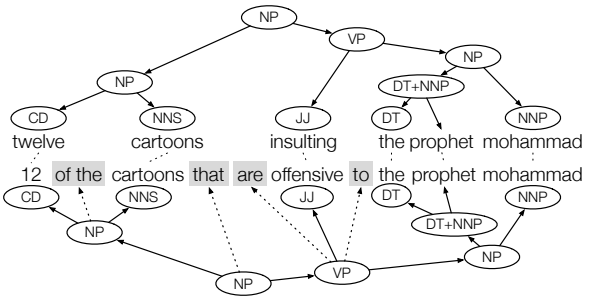

Figure 3: An example of a synchronous paraphrastic derivation, here a sentence compression. Shaded words are deleted in the indicated rule applications.

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

A wide variety of feature functions can be formulated. We detail the feature set used in our experiments in Section 6.2.

To obtain a paraphrase grammar, we first extract a translation grammar that translates a foreign language into English. 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 use the intuition that two English strings e_1 and e_2 that translate to the same foreign string f are equivalent in meaning, and *pivot* over f to create a paraphrase rule (Ganitkevitch et al., 2011; Callison-Burch, 2008; Bannard and Callison-Burch, 2005):

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

with a combined nonterminal correspondency function $\sim_p$. Similarly, 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:

$$p(e_2|e_1) = \sum_f p(e_2, f|e_1) \quad (2)$$

$$= \sum_f p(e_2|f, e_1)p(f|e_1) \quad (3)$$

$$\approx \sum_f p(e_2|f)p(f|e_1). \quad (4)$$

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 straightforwardly used within the normal machine translation machinery. Figure 3 shows an example for a synchronous paraphrastic derivation produced as a result of applying our grammar in the decoding process.

The approach we outlined in this section 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

Paraphrase extraction from monolingual corpora measures the similarity of phrases based on contextual features. To describe a phrase e, we define a set of features that describe the context of an occurrence of e in our corpus. The resulting feature vectors $\vec{s}_{e,i}$ are aggregated 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 identify 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 relie on dependency and constituency parses, part-of-speech tags, or lemmatization have been proposed early on (Church and Hanks, 1991; Lin and Pantel, 2001). For instance, in Lin and Pantel (2001)'s work, a phrase is described by the various syntactic relations it has with lexical items in its context, such as: "for

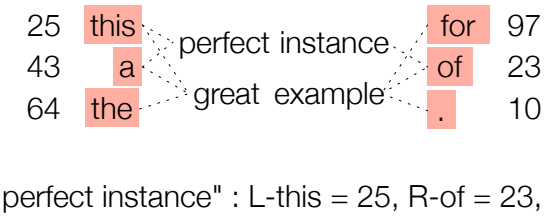

Figure 4: An example of immediate lexical context acquisition over n-gram corpora. Left- and right-adjacent unigrams are stored as features for the phrase. Here, "perfect instance" and "great example" share a number of contexts and would thus be assumed to be paraphrases.

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, i.e. "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 features for a variety of purposes in NLP and text-to-text generation. (Lapata and Keller, 2005; Bhagat and Ravichandran, 2008; Lin et al., 2010; Van Durme and Lall, 2010).

In this work, we will qualitatively and quantitatively compare the two approaches to distributional signature construction. Section 5 discusses the difference in effects on paraphrase ranking, and Section 6 presents human evaluation results on a text-to-text task.

2.3 Additional Related Work

In a GEMS workshop paper, Chan et al. (2011) presented an initial investigation into combining phrasal paraphrases obtained through bilingual pivoting with monolingual distributional information. They presented a re-ranking approach and evaluated their method via a substitution task. We expand on their by both substantially generalizing the formalism used, and by applying an evaluating our paraphrase system on a text-to-text generation task.

Figure 5: An example of word-labeled dependency-based rich distributional features over a parsed text corpus. The phrase "perfect instance" is annotated with the in- and outgoing dependency links, the words at the end of the links and a count.

3 Distributional Similarity Model

In order to investigate the impact of the feature set used, we chose to extract two collections of distributional similarity-based paraphrases. Using a web-scale n-gram corpus (Brants and Franz, 2006; Lin et al., 2010), we extract unigram features for the words to the left and right for phrases up to a length of 4. The features are weighed with the n-gram count given by the dataset. The resulting collection comprised context vectors for the 200 million most frequent 1- to 4-grams in the dataset.

For contrast, we use the constituency- and dependency-parsed Los Angeles Times/Washington Post portion of the Gigaword corpus (Graff et al., 2003). The following feature set is used to compute phrase contexts over this dataset:

- 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 is syntactically well-formed we additionally include lexical and POS features for its head.
- Syntactic features for constituents governing the phrase, as well as for CCG-style slashed constituent labels for the phrase, split by governing constituent and missing constituent.

Figure 5 illustrates our choice of feature set. As a result we obtain context information for over 12 million 1- to 4-gram phrases.

Much like Ravichandran et al. (2005) and Bhagat and Ravichandran (2008), we relied 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 in (Van Durme and Lall, 2010). 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 approximate the cosine similarity of u and v as:

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

where $D()$ is the Hamming distance.

4 Incorporating Distributional Similarity

In order to incorporate the 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 $\bar{e}_1$ and $\bar{e}_2$ this is a simple task, and the similarity score $sim(\bar{e}_1, \bar{e}_2)$ can be included in the rule's feature vector $\vec{\varphi}$. For rules whose right-hand sides contain gaps, computing a similarity score is less straightforward.

Figure 6 shows an example of a discontinuous rule and illustrates our solution: we decompose the discontinuous patterns that make up the right-hand sides of a rule $\mathbf{r}$ into pairs of contiguous phrases $\mathcal{P}(\mathbf{r}) = (\bar{e}, \bar{e}')$, 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 $\mathbf{a}$. We define and compute $\mathbf{a}$ analogously to the word alignments used in statistical machine translation.

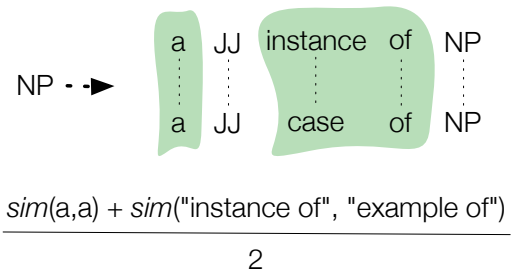

Figure 6: Scoring a rule by extracting and scoring contiguous phrases consistent with the alignment. The overall score of the rule is determined by

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 subphrases that are more likely to have been observed often enough for reliable signature and thus similarity estimates. The intended sparsity-reduction is similar to the backing-off concept in statistical n-gram language modeling.

Our two similarity scores are incorporated into the paraphraser as an additional rule feature φ_{sim}. The corresponding weight λ_{sim} is estimated along with the other λ_i in our training framework, detailed in Section 6.2.

5 Paraphrase Ranking

Finding good estimates of the quality of a paraphrase pair is crucial to the usefulness of a paraphrase system in practice. The key motivation to this work is that by combining information from both pivoted and monolingual distributional similarity-based paraphrases we can the quality of our system beyond either individual approach.

To test our hypothesis, we empirically analyze the effects the incorporation of distributional similarity has on the ranking of paraphrases within our grammar. In this section, we contrast the pivoting-based baseline system, with the integrations of the two previously described distributional similarity models: simple unigram features trained on a large n-gram corpus (which we will refer to as the "n-gram" model), and a similarity model with a richer feature set, but with significantly less coverage (dubbed "rich").

The ranking of paraphrase candidates is crucially dependent on the log-linear model weights λ_i. To make our analysis realistic, we adopt the feature weights produced by our tuning approach (see Section 6.2). In our rank calculation, we omit all compression-oriented features (such as word and character counts etc.), as well as the identity rule feature. This is done so that our ranking comparison is based on general paraphrase quality rather than task-specific utility.

Table 1 shows two example patterns pulled from our grammar, along with the top 5 paraphrases according to each system. Comparing across the systems, we notice that the baseline system appears to be biased towards the identity paraphrase. This is not surprising, as we would intuitively expect an English expression to pivot back onto itself more often than it would onto other synonymous expressions. While the inclusion of the n-gram distributional model doesn't seem to have any impact, the rich model noticeably flattens the distribution over candidates[1]. This suggests that by adding rich distributional similarity features, we can enable our system to paraphrase more freely and thus make it more flexible and expressive.

In addition to this, we can observe a number of high-ranking bad paraphrases such as *assume that* going to *is* or *sign of* paraphrasing as *with* that occur in the baseline system. These seem to be the result of errors in word alignment, and only in fringe cases would they result in grammatical or meaning-preserving output. While the n-gram model partially fixes the issue, it is limited by its unigram feature set, making it prone to errors such as reducing *sign of* to *of*. Again, the rich model fares significantly better here, mostly eliminating bad paraphrases in favor of true, and occasionally interesting paraphrases such as *imagine* for *assume that*.

Our observations suggest that a rich feature set is

[1] It should be noted, however, that the log-linear model costs the paraphrases are ranked by are only comparable within each column.

ADJP/NP → *ADJP/VP* assume that *S/NP*					
Pivot-Based		n-gram		Rich	
... assume that ...	8.39	... assume that ...	3.67	... imagine ...	1.99
... cause ...	8.88	... allow ...	4.02	... show ...	2.05
... show ...	8.93	... find ...	4.77	... to make ...	2.23
... is ...	9.04	... make ...	4.81	... cause ...	2.35
... make ...	9.11	... enable ...	4.89	... say ...	2.42

NP/PP → *DT* sign of *NP/PP*					
Pivot-Based		n-gram		Rich	
... sign of ...	8.98	... sign of ...	3.40	... view to ...	2.08
... with ...	10.25	... of ...	4.10	... attempt at ...	2.19
... view to ...	10.67	... with ...	5.50	... body of ...	2.46
... attempt at ...	10.80	... expression of ...	6.40	... face of ...	2.46
... face of ...	10.82	... setting of ...	6.63	... talk about ...	2.58

Table 1: Comparison of top 5 paraphrase rankings for two example patterns in our grammar. The top row shows the left-hand side and source side of the rule. The nonterminal symbols on the target side are identical to the source side in all cases and are omitted for legibility. The numbers next to the paraphrase candidates are the log-linear model costs, i.e. lower is better.

indeed valuable when integrating distributional similarity information into a pivot-based paraphrasing system. In the following, we will quantify this gain by evaluating our system on a text-to-text generation problem.

6 Experiments

6.1 Sentence Compression

To evaluate our method on a real text-to-text application, we chose to adopt the setting and datasets from (Ganitkevitch et al., 2011). We train our paraphrasing system to produce sentence level compression by way of sentential paraphrasing. We contrast our distributional similarity-informed paraphrase system with an uninformed pivoting-only (but otherwise identical) 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.

6.2 Experimental Setup

We extracted our paraphrase grammar from the French–English portion of the Europarl corpus (version 5) (Koehn, 2005). The Berkeley aligner and the Berkeley parser were used to align the bitext and parse the English side, respectively. The paraphrase grammar was produced using the Hadoop-based Thrax grammar extractor's paraphrase mode. 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 extended the feature set used by 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 c_{src} and c_{tgt} and the word count difference feature c_{diff}, we add character based count and difference features $char_{src}$, $char_{tgt}$, and $char_{diff}$, as well as log-compression ratio features $c_{cr} = \log \frac{c_{tgt}}{c_{src}}$ and the analogously defined $char_{cr}$.

For model tuning and decoding, we used the Joshua machine translation system (Weese et al., 2011). The model weights were 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).

6.3 Evaluation Results

To rate the quality of our output, we solicit human judgments of the compressions along two five-point scales: grammaticality and meaning. 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, and a grammar score of 1 is assigned to sentences that are entirely ungrammatical. It is known that evaluation quality correlates linearly with compression rate (Napoles et al., 2011). Thus, to ensure fairness in comparing our systems, we adjust compression rates to closely match on the sentence-level.

In Table 2 we compare our distributional similarity-augmented paraphraser to the plain pivoting baseline and the ILP approach at a compression rate of ≈ 0.79. We can see that the paraphrase approach significantly outperforms ILP on meaning retention. However it shows notable weaknesses in grammaticality. Adding n-gram-based distributional similarity information to the paraphrases recovers some of the difference in grammaticality and also yields gain in the compressions' meaning retention. Moving to the rich distributional signatures yields additional improvement.

Figure 7 shows a pairwise comparison breakdown detailing the number of wins and ties in the human judgements for each comparison. As suggested by our analysis in Section 5, there is substantial overlap between the baseline system and the n-gram distributional similarity model, while the rich feature set leads to noticeably different output far more often. The meaning retention improvements over both the baseline and ILP are statistically significant at $p < 0.05$.

Table 3 shows an example sentence drawn from our test set and the compressions produced by the different systems. We see that both the paraphrase and ILP systems produce good quality results, with the paraphrase system retaining the meaning of the source sentence more accurately.

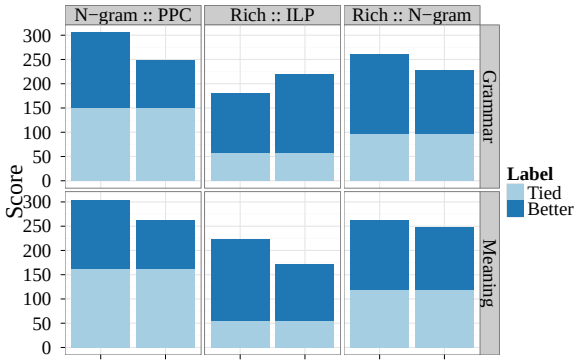

Figure 7: A breakdown of the of the collected human judgments comparing the systems head to head.

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

Table 2: Results of the human evaluation on longer compressions: pairwise compression rates (CR), meaning and grammaticality scores. Bold indicates a statistically significance difference at $p < 0.05$.

7 Conclusion

We presented a method to incorporate monolingual distributional similarity into linguistically informed sentential paraphrase extracted from bilingual parallel data. We investigated both the effect of varying the feature set used for determining the distributional signatures of phrases and conclude that a richer feature set, even with significantly lower coverage, seems to noticeably improve paraphrase quality in ranking. We evaluated our integrated paraphrases on a text-to-text generation task and show that our method significantly improves meaning retention over both a strong paraphrastic baseline and a specialized state-of-the-art system.

Source	should these political developments have an impact on sports ?
Reference	should these political events affect sports ?
Rich	should these events have an impact on sports ?
n-gram	these political developments impact on sports ?
PC	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 .
Rich	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 .
PC	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
Source	we are not poor . in fact , in our country , we are rich , and our poverty is in the lack of patriotism by some of the officials among us .
Reference	we are not poor in our country ; we are rich . our poverty is in the lack of patriotism of some of our officials .
Rich	we are not poor . in fact in our country we are rich , and our poverty is the lack of patriotism by some officials us .
n-gram	we are poor . in fact , our country , we are rich and poverty is no patriotism by some officials us .
PC	we are not poor . in fact , in our country rich , and poverty is no patriotism by some officials among us .
ILP	we are not poor in fact in our country we are rich and our poverty is in lack by some of officials among us

Table 3: Example compressions produced by our systems and the baselines Table 2 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*.

T.P. Chan, C. Callison-Burch, and B. 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, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614.

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*.

D. Graff, J. Kong, K. Chen, and K. Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

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*.

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

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

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*.

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

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*.

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

D. Ravichandran, P. Pantel, and E. 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*.

J. Weese, J. Ganitkevitch, C. Callison-Burch, M. Post, and A. Lopez. 2011. Joshua 3.0: Syntax-based machine translation with the Thrax grammar extractor. In *Proceedings of WMT11*.