Sentential Paraphrasing as Black-Box Machine Translation

Courtney Napoles[1], **Chris Callison-Burch**[2], and **Matt Post**[3]
[1] Center for Language and Speech Processing, Johns Hopkins University
[2] Computer and Information Science Department, University of Pennsylvania
[3] Human Language Technology Center of Excellence, Johns Hopkins University

Abstract

We present a simple, prepackaged solution to generating paraphrases of English sentences. We use the Paraphrase Database (PPDB) for monolingual sentence rewriting and provide machine translation *language packs:* prepackaged, tuned models that can be downloaded and used to generate paraphrases on a standard Unix environment. The language packs can be treated as a black box or customized to specific tasks. In this demonstration, we will explain how to use the included interactive web-based tool to generate sentential paraphrases.

1 Introduction

Monolingual sentence rewriting encompasses a variety of tasks for which the goal is to generate an output sentence with similar meaning to an input sentence, in the same language. The generated sentences can be called *sentential paraphrases*. Some tasks that generate sentential paraphrases include sentence simplification, compression, grammatical error correction, or expanding multiple reference sets for machine translation. For researchers not focused on these tasks, it can be difficult to develop a one-off system due to resource requirements.

To address this need, we are releasing a black box for generating sentential paraphrases: machine translation language packs. The language packs consist of prepackaged models for the Joshua 6 decoder (Post et al., 2015) and a monolingual "translation" grammar derived from the Paraphrase Database (PPDB) 2.0 (Pavlick et al., 2015). The PPDB provides tremendous coverage over English text, containing more than 200 million paraphrases extracted from 100 million sentences (Ganitkevitch et al., 2013). For the first time, any researcher with Java 7 and Unix (there are no other dependencies) can generate sentential paraphrases without developing their own system. Additionally, the language packs include a web tool for interactively paraphrasing sentences and adjusting the parameters.

The language packs contain everything needed to generate sentential paraphrases in English:

- a monolingual synchronous grammar,
- a language model,
- a ready-to-use configuration file,
- the Joshua 6 runtime, so that no compilation is necessary,
- a shell script to invoke the Joshua decoder, and
- a web tool for interactive decoding and parameter configuration.

The system is invoked by a single command, either on a batch of sentences or as an interactive server.

Users can choose which size grammar to include in the language pack, corresponding to the PPDB pack sizes (S through XXXL).

In the rest of the paper, we will describe the translation model and grammar, provide examples of output, and explain how the configuration can be adjusted for specific needs.

2 Language pack description

Several different size language packs are available for download.[1] The components of the language packs are described below.

[1] `http://joshua-decoder.com/language-packs/paraphrase/`

Grammar Our approach to sentential paraphrasing is analogous to machine translation. As a translation grammar, we use PPDB 2.0, which contains 170-million lexical, phrasal, and syntactic paraphrases (Pavlick et al., 2015). Each language pack contains a PPDB grammar that has been packed into a binary form for faster computation (Ganitkevitch et al., 2012), and users can select which size grammar to use. The rules present in each grammar are determined by the PPDB 2.0 score, which indicates the paraphrase quality (as given by a supervised regression model) and correlates strongly with human judgments of paraphrase appropriateness (Pavlick et al., 2015). Grammars of different sizes are created by changing the paraphrase score thresholds; larger grammars therefore contain a wider diversity of paraphrases, but with lower confidences.

Features Each paraphrase in PPDB 2.0 contains 44 features, described in Ganitkevitch and Callison-Burch (2014) and Pavlick et al. (2015). For each paraphrase pair, we call the input the *original* and the new phrase the *candidate*. Features can reflect just the candidate phrase or a relationship between the original and candidate phrases. Each of these features is assigned a weight, which guides the decoder's choice of paraphrases to apply to generate the final candidate sentence. All feature values are pre-calculated in PPDB 2.0.

Decoding The language packs include a compiled Joshua runtime for decoding, a script to invoke it, and configuration files for different tuned models. There is also a web-based tool for interactively querying a server version of the decoder for paraphrases. We include a 5-gram Gigaword v.5 language model for decoding. One or more language-model scores are used to rank translation candidates during decoding. The decoder outputs the *n*-best candidate paraphrases, ranked by model score.

3 Models

Each language pack has three pre-configured models to use either out of the box or as a starting point for further customization. There are tuned models for (1) sentence compression, (2) text simplification, and (3) a general-purpose model with hand-tuned weights. These models are distinguished only by the different weight vectors, and are selected by point the Joshua invocation script to the corresponding configuration file.

3.1 Tuned models

We include two models that were tuned for (1) sentence compression and (2) simplification. The compression model is based on the work of Ganitkevitch et al. (2011), and uses the same features, tuning data, and objective function, PRÉCIS. The simplification model is described in Xu et al. (2016), and is optimized to the SARI metric. The system was tuned using the parallel data described therein as well as the Newsela corpus (Xu et al., 2015). There is no specialized grammar for these models; instead, the parameters were tuned to choose appropriate paraphrases from the PPDB.

Sample output generated with these models is shown in Table 1.

3.2 Hand-derived weights

To configure the general-purpose model, which generates paraphrases for no specific task, we examined the output of 100 sentences randomly selected from each of three different domains: newswire (WSJ 0–1 (Marcus et al., 1993)), "simple" English (the Britannica Elementary corpus (Barzilay and Elhadad, 2003)), and general text (the WaCky corpus (Baroni et al., 2009)). We systematically varied the weights of the Gigaword LM and the PPDB 2.0 score features and selected values that yielded the best output as judged by the authors. The parameters selected for the generic language packs are $weight_{\texttt{lm}} = 10$ and $weight_{\texttt{ppdb2}} = 15$, with all other weights are set to zero. Example output is shown in Table 1.

4 User customization

The language packs include configuration files with pre-determined weights that can be used on their own or as a jumping-off point for custom configurations. There are weights for each of the 44 PPDB 2.0 features as well as for the language model(s) used by the decoder. We encourage researchers to explore modifications to the model to suit their specific tasks, and we have clearly identified five aspects of the language packs that can be modified:

1. Alternate language models. The decoder can accept multiple LMs, and the packs include LMs es-

Compression	
Orig:	rice admits mistakes have been made by american administration in rebuilding iraq
Gen:	rice admits mistakes were made by american administration in rebuilding iraq
Orig:	partisanship is regarded as a crime , and pluralism is rejected , and no one in the shura council would seek to compete with the ruler or distort his image .
Gen:	partisanship is regarded as a crime [] and pluralism is rejected [] and none in the shura council would seek to compete with the ruler or distort his image .
Simplification	
Orig:	fives is a british sport believed to derive from the same origins as many racquet sports .
Gen:	fives is a british sport thought to come from the same source as many racquet sports .
Orig:	in the soviet years , the bolsheviks demolished two of rostov 's principal landmarks — st alexander nevsky cathedral (1908) and st george cathedral in nakhichevan (1783-1807) .
Gen:	in the soviet years , the bolsheviks destroyed two of rostov 's key landmarks — st alexander nevsky church (1908) and st george church in naxçivan (1783-1807) .
Generic	
Orig:	because the spaniards had better weapons , cortes and his army took over tenochtitlan by 1521 .
Gen:	as the spaniards had better weapons , cortes and his men took over tenochtitlan by 1521 .
Orig:	it was eventually abandoned due to resistance from the population .
Gen:	it was later abandoned due to opposition from the population .

Table 1: Sample output from the three models. Underlines designate changed spans, and [] indicates deletions.

timated over newswire text and "simple" English. Other user-provided LMs can be used for tasks targeting different domains of text.

2. Rank output with a custom metric. The n-best candidate sentences are chosen by their score according to a given metric (LM score for the generic model, and PRÉCIS and SARI for the tuned models), however other metrics can be used instead.

3. Manually adjust parameters. The weights of the features discussed in Section 3 can be adjusted, as well as other PPDB feature weights. The web tool (Figure 1) allows users to select the weights for all of the features and see the top-5 candidates generated with those weights. Some of the more interpretable features to target include the length difference and entailment relations between the phrase original and candidate, as well as formality and complexity scores of the candidate paraphrase.

4. Optimize parameters with parallel data. For tailoring machine translation to a specific task, the weights given to each feature can be optimized to a given metric over a tuning set of parallel data. This metric is commonly BLEU in machine translation, but it can be a custom metric for a specific task, such as PRÉCIS for compression (Ganitkevitch et al., 2011) or SARI for simplification (Xu et al., 2016). The user needs to provide a parallel dataset for tuning, ideally with about 2,000 thousand sentences. The pipeline scripts in the Joshua decoder have options for optimization, with the user specifying the language pack grammar and parallel tuning data. The configuration file included in the language pack can be used as a template for tuning.

5 Interactive tool

Finally, we include a web tool that lets users interact with the decoder and choose custom weights (Figure 1). Once users have downloaded the tool kit, an included script lets them run the decoder as a server, and through the web interface they can type individual sentences and adjust model parameters. The interface includes an input text box (one sentence

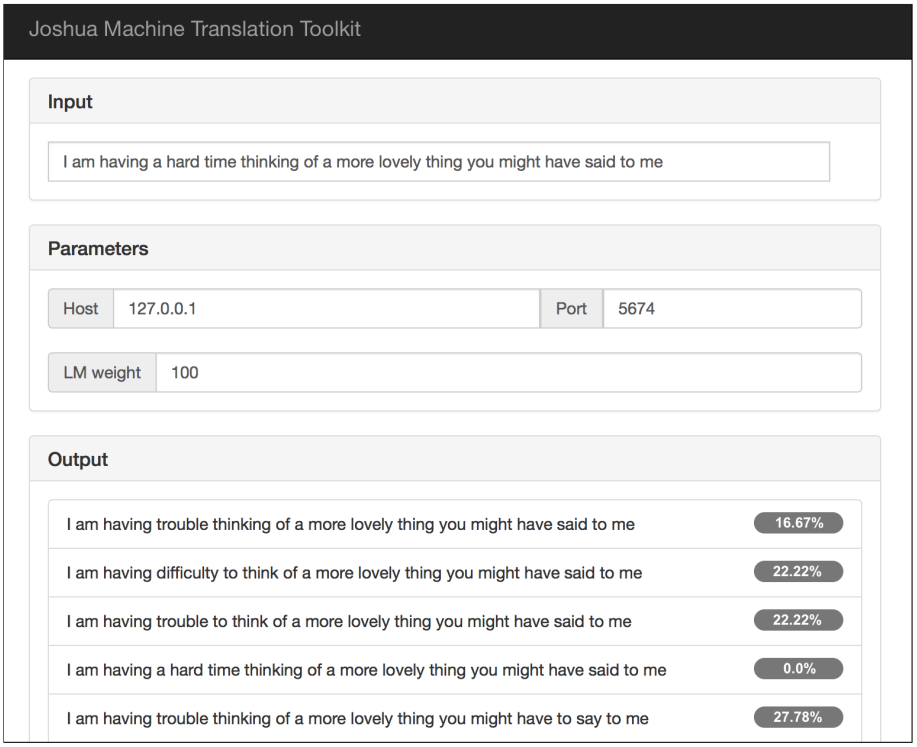

Figure 1: A screen shot of the web tool. The number to the right of each output sentence is the TER.

at a time), and slider bars to change the weights of any of the features used for decoding. Since this model has not been manually evaluated, we favor precision over recall and maintain a relatively conservative level of paraphrasing. The user is shown the top 10 outputs, as ranked by the sentence score. For each output sentence, we report the Translation Edit Rate (TER), which is the number of changes needed to transform the output sentence into the input (Snover et al., 2006).

This tool can be used to demonstrate and test a model or to hand-tune the model in order to determine the parameters for a configuration file to paraphrase a large batch of sentences. Detailed instructions for using the tool and shell scripts, as well as a detailed description of the configuration file, are available at the language pack home page: `http://joshua-decoder.com/language-packs/paraphrase/`

6 Related work

Previous work has applied machine translation techniques to monolingual sentence rewriting tasks. The most closely related works used a monolingual paraphrase grammar for sentence compression (Ganitkevitch et al., 2011) and sentence simplification (Xu et al., 2016), both of which developed custom metrics and task-specific features. Various other MT approaches have been used for generating sentence simplifications, however none of these used a general-purpose paraphrase grammar (Narayan and Gardent, 2014; Wubben et al., 2012, among others). Another application of sentential paraphrases is to expand multiple reference sets for machine translation (Madnani and Dorr, 2010).

PPDB has been used for many tasks, including recognizing textual entailment, question generation, and measuring semantic similarity.

These language packs were inspired by the foreign language packs released with Joshua 6 (Post et al., 2015).

7 Conclusion

We have presented a black box for generating sentential paraphrases: PPDB language packs. The language packs include everything necessary for generation, so that they can be downloaded and invoked with a single command. This toolkit can be used for

a variety of tasks: as a helpful tool for writing (what is another way to express a sentence?); generating additional training or tuning data, such as multiple-references for machine translation or other text-to-text rewriting tasks; or for changing the style or tone of a text. We hope their ease-of-use will facilitate future work on text-to-text rewriting tasks.

Acknowledgments

This material is based upon work partially supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1232825. This research was supported by the Human Language Technology Center of Excellence, and by gifts from the Alfred P. Sloan Foundation, Google, and Facebook. This material is based in part on research sponsored by the NSF grant under IIS1249516 and DARPA under number FA8750-13-20017 (the DEFT program). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA and the U.S. Government.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *LREC*, pages 4276–4283. Citeseer.

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 the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1168–1179, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post, and Chris Callison-Burch. 2012. Joshua 4.0: Packing, PRO, and Paraphrases. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 283–291, Montréal, Canada, June. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *HLT-NAACL*, pages 758–764.

Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, (Early Access):1–47.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 435–445.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July. Association for Computational Linguistics.

Matt Post, Yuan Cao, and Gaurav Kumar. 2015. Joshua 6: A phrase-based and hierarchical statistical machine translation system. *The Prague Bulletin of Mathematical Linguistics*, 104(1):5–16.

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.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4.