N-gram Counts and Language Models from the Common Crawl

Christian Buck[†], **Kenneth Heafield**[‡], **Bas van Ooyen**[*]
[†]University of Edinburgh, Edinburgh, Scotland
[‡]Stanford University, Stanford, CA, USA
[*] Owlin BV, Utrecht, Netherlands
christian.buck@ed.ac.uk, heafield@cs.stanford.edu, bas@owlin.com

Abstract

We contribute 5-gram counts and language models trained on the Common Crawl corpus, a collection over 9 billion web pages. This release improves upon the Google n-gram counts in two key ways: the inclusion of low-count entries and deduplication to reduce boilerplate. By preserving singletons, we were able to use Kneser-Ney smoothing to build large language models. This paper describes how the corpus was processed with emphasis on the problems that arise in working with data at this scale. Our unpruned Kneser-Ney English 5-gram language model, built on 975 billion deduplicated tokens, contains over 500 billion unique n-grams. We show gains of 0.5–1.4 BLEU by using large language models to translate into various languages.

Keywords: web corpora, language models, multilingual

1. Introduction

The sheer amount of data in multiple languages makes web-scale corpora attractive for many natural language processing tasks. Of particular importance is language modeling, where web-scale language models have been shown to improve machine translation and automatic speech recognition performance (Brants et al., 2007; Chelba and Schalkwyk, 2013; Guthrie and Hepple, 2010). In this work, we contribute n-gram counts and language models trained on the Common Crawl corpus.[1]

Google has released n-gram counts (Brants and Franz, 2006) trained on one trillion tokens of text. However, they pruned any n-grams that appeard less than 40 times. Moreover, all words that appeared less than 200 times were replaced with the unknown word. Both forms of pruning make the counts unsuitable for estimating a language model with the popular and successful Kneser-Ney smoothing algorithm, which requires unpruned counts even if the final model is to be pruned.

The second issue with the publicly available Google n-gram counts (Brants and Franz, 2006) is that the training data was not deduplicated, so boilerplate such as copyright notices has unreasonably high counts (Lin et al., 2010). Google has shared a deduplicated version (Bergsma et al., 2010) in limited contexts (Lin et al., 2010), but it was never publicly released (Lin, 2013). Our training data was deduplicated before counting n-grams.

Microsoft provides a web service (Wang et al., 2010) that can be queried for language model probabilities. The service is currently limited to the English language whereas we provide models for many languages. Moreover, an initial experiment on reranking English machine translation output led to so many queries that the service went down several times, despite client-side caching. Using the Microsoft service during machine translation decoding would entail far more queries and require lower latency.

2. Data Preparation

The Common Crawl[2] is a publicly available crawl of the web. We use the 2012, early 2013, and "winter" 2013 crawls, consisting of 3.8 billion, 2 billion, and 2.3 billion pages, respectively. Because both 2013 crawls are similar in terms of seed addresses and distribution of top-level domains in this work we only distinguish 2012 and 2013 crawls.

The data is made available both as raw HTML and as text only files. The latter collection consists of all HTML and RSS files from which all tags were stripped. The HTML comes in the original encoding, while the text has been converted to UTF-8, albeit with the occasional invalid character.

Using the HTML files has the advantage of being able to exploit the document structure to select paragraphs and to tell boilerplate from actual content. However, parsing such large amounts of HTML is non-trivial and requires many normalization steps.

In this work we focus on processing the text only files which we downloaded and processed locally on a small cluster. The advantages of structured text do not outweigh the extra computing power needed to process them.

2.1. Language Detection

The first step in our pipeline is splitting the data by language. We explored the option of automatically detecting the main language for every page but found that mixed-language content is quite common. By using the Compact Language Detector 2 (CLD2)[3] we are able to partition every document into monolingual spans. CLD2 is able to detect 175 languages and fast enough to process the entire corpus within a week.

Table 1 shows the relative contribution of the most common languages in the separated data. At this stage of processing we have no meaningful notion of token or line counts and therefore report the size of the extracted files. As

[1] http://statmt.org/ngrams

[2] http://commoncrawl.org/

[3] https://code.google.com/p/cld2/

Language	Relative occurrence % 2012	2013	both	Size both
English	54.79	79.53	67.05	23.62 TiB
German	4.53	1.23	2.89	1.02 TiB
Spanish	3.91	1.68	2.80	986.86 GiB
French	4.01	1.14	2.59	912.16 GiB
Japanese	3.11	0.14	1.64	577.14 GiB
Russian	2.93	0.09	1.53	537.36 GiB
Polish	1.81	0.08	0.95	334.31 GiB
Italian	1.40	0.44	0.92	325.58 GiB
Portuguese	1.32	0.48	0.90	316.87 GiB
Chinese	1.45	0.04	0.75	264.91 GiB
Dutch	0.95	0.22	0.59	207.90 GiB
other	12.23	12.57	12.40	4.37 TiB

Table 1: Results of language detection on raw text, showing the 11 most common languages.

expected English is by far the predominant language followed by European languages. Due to the large size of the overall corpus (35.23 TiB), even a small percentage constitutes a corpus of useful size. For example, only 0.14% of the data were classified as Finnish, yet yielding a corpus of 47.73 GiB. In total we found 73 languages with at least 1 GiB of uncompressed text each and 42 with at least 10 GiB.

2.2. Deduplication

Since the Common Crawl contains web pages, many fragments are not content but artifacts of automatic page generation, such as copyright notices. In order to reduce the amount of boilerplate, we remove duplicate lines prior to sentence splitting. While a selection of very common lines in Table 2 suggests that mostly irrelevant data is removed, we do risk deduplicating content that *should* appear repeatedly.

Storing all of the lines in memory would take too much RAM. Instead, we take a 64-bit hash of each line using MurmurHash[4]. If the hash was not seen before we keep the line and add it to an in-memory hash-table. However, for common languages, such as English, it is not feasible to keep all hashes in memory. We therefore shard the data using a different hash, so that all identical lines end up in the same shard. We then deduplicate each shard individiually. While hash collisions can lead to lines being incorrectly identified as duplicates we found that on a 10 GiB sample of the corpus no such errors were made.

The deduplication step removes about 80% of the English data which is in line with the reductions reported by Bergsma et al. (2010). Comparing Table 1 and Table 3 we find that this rate is lower for other languages, e.g. about 2/3 for Spanish and German. We speculate that this is due to mixed language content on websites, where boilerplate text may be English despite the main content appearing in another language.

Count (M)	Line
1374.44	Add to
816.33	Share
711.68	Unblock User
68.31	Sign in or sign up now!
61.26	Log in
54.77	Privacy Policy
45.18	April 2010
34.35	Load more suggestions
19.84	Buy It Now | Add to watch list
16.64	Powered by WordPress.com

Table 2: Selection of very common lines in the English portion of the data. We only keep one instance. The counts are given as million lines.

Language	Lines (B)	Tokens (B)	Bytes
English	59.13	975.63	5.14 TiB
German	3.87	51.93	317.46 GiB
Spanish	3.50	62.21	337.16 GiB
French	3.04	49.31	273.96 GiB
Russian	1.79	21.41	220.62 GiB
Czech	0.47	5.79	34.67 GiB

Table 3: Data statistics after preprocessing

2.3. Normalization

In addition to deduplicating, we restricted the data to printable Unicode characters, replaced all e-mail addresses with the same address, stripped out remaining HTML, and split sentences using the Europarl splitter (Koehn, 2005). We distribute the text after this stage for those who wish to use their own tokenizer.

Before building language models, we normalized punctuation using the script provided by the Workshop on Statistical Machine Translation (Bojar et al., 2013), tokenized using the Moses tokenizer (Koehn et al., 2007), and applied the Moses truecaser. The truecaser was trained on data from the 2014 Workshop on Statistical Machine Translation for each language, including Europarl (Koehn, 2005), United Nations parallel data, the Giga Fr-En corpus, parallel data mined from CommonCrawl (Smith et al., 2013), the news commentary corpus, LDC Gigaword corpora (Parker et al., 2011), and the news crawls provided by the evaluation. Our release includes a frozen version of the scripts and truecasing model used to preprocess the data.

Table 3 gives some statistics of the data after all preprocessing steps have been performed.

3. Language Model Estimation

After preprocessing, we are left with several large monolingual corpora. By using disk-based streaming (Heafield et al., 2013) we are able to efficiently estimate language models much larger than the physical memory on our machines. For example, estimating a language model on 535 billion tokens took 8.2 days a single machine with 140 GiB

[4] https://sites.google.com/site/murmurhash/

n-gram length	count
1	2 640 258 088
2	15 297 753 348
3	61 858 786 129
4	156 775 272 110
5	263 690 452 834

Table 4: N-gram counts for the English language model.

RAM. For all languages for which we have sufficient data and a preprocessing pipeline, we produce unpruned 5-gram models using interpolated modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998).

We don't give details for all models but the largest one. Table 4 shows n-gram counts for the English language model that was estimated on almost a trillion tokens. The resulting model has a size of 5.6TB.

4. Experiments

To evaluate the language models, we compute perplexity and run machine translation experiments. Measurements are based on the 3003-sentence test set from the 2014 Workshop on Statistical Machine Translation shared task, with encoding issues fixed by the organizers. Since Spanish was not part of the 2014 evaluation, we use the 2013 test set. These test sets, hereafter referred to as *newstest*, were drawn from news articles in various languages and were professionally translated to the other languages. One issue that arises when using data collected online is the possibility that sentences from the test set might occur in the training data. We use the most recent test data to minimize the possible overlap.

4.1. Perplexity

Tables 5, 6, and 7 show perplexity results on the newstest corpus. We report perplexity both by skipping unknown words and by including unknown words with $p(\text{<unk>})$. For comparison, we also computed perplexity using language models trained on various corpora allowed by the constrained condition of the 2014 Workshop on Statistical Machine Translation. We preprocessed the training data and newstest data using the same normalization, tokenization, and truecasing steps described in Section 2.3. The monolingual corpora from which these models were estimated are much smaller but are generally cleaner. Because the duplication rate was low, we did not deduplicate training data for constrastive models, though some corpora were already deduplicated beforehand.

A common problem with perplexity comparisons is that models have different vocabularies. For example, a language model could achieve low perplexity with a small vocabulary and a high probability for the unknown word. To work around this problem, we took the maximum vocabulary size and applied it to all models. Essentially, we ensure that all words appear in all models, even if they have count zero in a given model's training data. In Kneser-Ney smoothing, these count-zero words act just like copies of the unknown word. Their probability mass arises when the unigrams are interpolated with the uniform distribution, adding

$$\frac{1}{|\text{vocabulary}|}$$

times the interpolation weight to each unigram probability. We simply use the same vocabulary size in the denominator of this equation for all models being compared. As a result, all models sum to 1 over the entire vocabulary. We emphasize that this approach is not new, but rather standard practice recommended by IRSTLM (Federico et al., 2008). The tables show perplexity results with models trained with a consistent vocabulary size. However, we also trained models in the normal way and found that the large language model still had smaller perplexity, despite having the smallest unknown word probability.

		Perplexity		
Corpus		skip	include	OOVs
Europarl		357.99	620.58	1902
United Nations		378.83	484.47	863
Giga Fr-En (English)		273.57	303.08	355
Common Crawl parallel		266.86	299.43	418
News Commentary		349.39	696.20	2568
English Gigaword	afp	171.72	190.82	346
	apw	166.83	185.62	344
	cna	308.86	498.88	1713
	ltw	177.69	215.28	626
	wpb	229.88	334.74	1341
	nyt	161.12	179.22	338
	xin	205.91	238.50	499
News	2007	176.06	204.58	517
	2008	147.28	161.68	311
	2009	142.59	158.32	346
	2010	153.34	172.41	394
	2011	137.27	149.38	275
	2012	129.85	139.59	235
	2013	109.52	113.74	122
All interpolated		92.69	93.81	46
This work		**58.44**	**58.55**	**5**

Table 5: Perplexities on English newstest 2014. The "skip" column ignores unknown words for purposes of perplexity computation while the "include" column uses $p(\text{<unk>})$. Since all models were trained with the same vocabulary size, the "include" column is more directly comparable across corpora.

4.2. Machine Translation

For a practical application of the models presented in this work we add them to a Machine Translation system. For this we did not build new models ourselves but used those that were produced for the WMT 2014 Machine Translation shared task.

The baseline systems were trained using Moses (Koehn et al., 2007) with the following features: maximum sentence length of 80, grow-diag-final-and symmetrization of

		Perplexity		
Corpus		skip	include	OOVs
Europarl		180.73	269.28	1747
United Nations		189.90	230.88	861
Giga Fr-En (French)		143.77	156.56	372
Common Crawl parallel		143.36	157.84	446
News Commentary		186.08	318.83	2558
French Gigaword	afp	91.62	99.06	320
	apw	105.98	120.21	539
News	2007	173.44	268.26	2183
	2008	99.55	111.04	474
	2009	100.14	111.47	468
	2010	110.17	125.87	596
	2011	89.02	96.84	361
	2012	89.74	97.67	365
	2013	74.37	78.62	235
All interpolated		**60.78**	**62.02**	88
This work		65.50	65.76	**16**

Table 6: Perplexities on French newstest 2014. Interpolating the cleaner data led to lower perplexity than using CommonCrawl alone.

		Perplexity		
Corpus		skip	include	OOVs
Europarl		219.51	327.38	1596
United Nations		253.37	327.21	1004
Common Crawl parallel		191.48	229.49	766
News Commentary		225.26	374.89	2202
Spanish Gigaword	afp	152.73	173.04	480
	apw	173.51	203.23	615
	xin	195.15	234.58	741
News	2007	245.23	507.61	3386
	2008	171.18	209.14	815
	2009	173.04	216.08	916
	2010	183.58	239.15	1109
	2011	139.65	161.30	561
	2012	140.91	163.51	584
This work		**99.08**	**99.38**	**124**

Table 7: Perplexities on Spanish newstest 2013

	BLEU		
Language Pair	Baseline	+ this work	Δ
English-Czech	21.5	22.1	0.6
English-Hindi	11.1	12.5	1.4
English-Russian	28.7	29.9	1.2
English-German	20.5	21.0	0.5
Hindi-English	15.3	16.2	0.9

Table 8: Results of adding the language models presented in this work to an MT system. The given results refer to uncased BLEU scores.

	BLEU			
	2012	Δ	2013	Δ
Baseline	35.8		30.9	
+ 50M lines	36.3	0.5	31.5	0.6
+ 100M lines	36.5	0.7	31.5	0.6
+ 200M lines	36.6	0.8	31.8	0.9
+ 400M lines	37.0	1.2	31.8	0.9
+ 800M lines	37.3	1.6	31.8	0.9
+ 1.3B lines	37.7	1.9	32.0	1.1

Table 9: Machine Translation performance for English-Spanish on newstest 2012/2013 using increasing amounts of data for the additional language model.

GIZA++ alignments, an interpolated Kneser-Ney smoothed 5-gram language model with KenLM (Heafield et al., 2013) used at runtime, a lexically-driven 5-gram operation sequence model (Durrani et al., 2013), msd-bidirectional-fe lexicalized reordering, sparse lexical and domain features (Hasler et al., 2012), a distortion limit of 6, 100-best translation options, Minimum Bayes Risk decoding (Kumar and Byrne, 2004), Cube Pruning (Huang and Chiang, 2007), with a stack-size of 1000 during tuning and 5000 during test and the no-reordering-over-punctuation heuristic. The English-to-German systems also use POS and morphological target sequence models built on the in-domain subset of the parallel corpus using Kneser-Ney smoothed 7-gram models and as additional factors in phrase translation models (Koehn and Hoang, 2007). Additionally target-side language models over automatically built word-classes (Birch et al., 2013) were built. The Hindi-English system uses transliteration for unknown words (Durrani et al., 2014).

These results[5] are based on automatic BLEU (Papineni et al., 2002) scores as human evaluations were not available at time of writing. The baseline systems performed well in the shared task as measured by BLEU yielding an improvement between 0.5% and 1.4% over the baseline as shown in Table 8.

Finally, we investigate the relation between the amount of Common Crawl data used and improvements in MT quality. To this end we train a system using Moses and standard settings but all available parallel data as detailed in Section 2.3. with the exception of 2013 news data. Next, we add a language model trained on a sample of the available data and retune the system. The samples are selected such that each larger sample is a superset of any smaller one. Results in Table 9 show that even though the web data is quite noisy even limited amounts give improvements. We should however keep in mind that these numbers may be optimistic as we cannot rule out the possibility that some of the segments appear in the Common Crawl data.

[5] http://matrix.statmt.org/

5. Conclusion

We release n-gram counts and language models built on very large corpora which overcome limitations of similar publicly available resources. We show that even without sophisticated cleaning of the data we obtain results that outperform state-of-the-art language models used in Statistical Machine Translation. We show that improvements in perplexity also lead to better translations when used during decoding.

Acknowledgements

The authors would like to thank Nadir Durrani for the baseline models used in Section 4.2.
This work was supported by the MateCat project, which is funded by the EC under the 7^{th} Framework Programme.
This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Specifically, we used Stampede provided by the Texas Advanced Computing Center (TACC) at The University of Texas at Austin under NSF XSEDE allocation TG-CCR140009. We also acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Broad Operational Language Translation (BOLT) program through IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA or the US government.

6. References

Shane Bergsma, Emily Pitler, and Dekang Lin. 2010. Creating robust supervised classifiers via web-scale n-gram data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 865–874. Association for Computational Linguistics, July.

Alexandra Birch, Nadir Durrani, and Philipp Koehn. 2013. Edinburgh SLT and MT System Description for the IWSLT 2013 Evaluation. In *Proceedings of the 10th International Workshop on Spoken Language Translation*, pages 40–48, Heidelberg, Germany, December.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Thorsten Brants and Alex Franz. 2006. The Google web 1T 5-gram corpus version 1.1. LDC2006T13.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, pages 858–867, June.

Ciprian Chelba and Johan Schalkwyk, 2013. *Empirical Exploration of Language Modeling for the google.com Query Stream as Applied to Mobile Voice Search*, pages 197–229. Springer, New York.

Stanley Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, August.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In *Proceedings of the 15th Conference of the European Chapter of the ACL (EACL 2014)*, Gothenburg, Sweden, April. Association for Computational Linguistics. To appear.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech*, Brisbane, Australia.

David Guthrie and Mark Hepple. 2010. Storing the web in memory: Space efficient language models with constant time retrieval. In *Proceedings of EMNLP 2010*, Los Angeles, CA.

Eva Hasler, Barry Haddow, and Philipp Koehn. 2012. Sparse Lexicalised features and Topic Adaptation for SMT. In *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 268–275.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic, June. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184.

Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June. Association for Computational Linguistics.

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 *Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, June.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Shankar Kumar and William J. Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *HLT-NAACL*, pages 169–176.

Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma andKailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. 2010. Final report of the 2009 JHU CLSP workshop, June.

Dekang Lin. 2013. Personal communication, October.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, July.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, June. LDC2011T07.

Jason Smith, Hervé Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of ACL*. Association for Computational Linguistics, August.

Kuansan Wang, Christopher Thrasher, Evelyne Viegas, Xiaolong Li, and Bo-june Paul Hsu. 2010. An overview of Microsoft web n-gram corpus and applications. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 45–48. Association for Computational Linguistics.