Annotated Gigaword

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme
Human Language Technology Center of Excellence
Johns Hopkins University

Abstract

We have created layers of annotation on the English Gigaword v.5 corpus to render it useful as a standardized corpus for knowledge extraction and distributional semantics. Most existing large-scale work is based on inconsistent corpora which often have needed to be re-annotated by research teams independently, each time introducing biases that manifest as results that are only comparable at a high level. We provide to the community a public reference set based on current state-of-the-art syntactic analysis and coreference resolution, along with an interface for programmatic access. Our goal is to enable broader involvement in large-scale knowledge-acquisition efforts by researchers that otherwise may not have had the ability to produce such a resource on their own.

1 Introduction

Gigaword is currently the largest static corpus of English news documents available. The most recent addition, Gigaword v.5 (Parker et al., 2011), contains nearly 10-million documents from seven news outlets, with a total of more than 4-billion words. We have annotated this collection with syntactic and discourse structure, for release to the community through the Linguistic Data Consortium (LDC) as a static, large-scale resource for knowledge acquisition and computational semantics. This resource will (1) provide a consistent dataset of state-of-the-art annotations, over which researchers can compare results, (2) prevent the reduplication of annotation efforts by different research groups, and (3) "even the playing field" by better enabling those lacking the computational capacity to generate such annotations at this scale.

The Brown Laboratory for Linguistic Information Processing (BLLIP) corpus (Charniak et al., 2000) contains approximately 30-million words of Wall Street Journal text, annotated with automatically derived Treebank-style parses and part-of-speech tags. This was followed by the BLLIP North American News Text corpus (McClosky et al., 2008), containing approximately 350-million words of syntactically parsed newswire.

Through the Web-as-Corpus kool ynitiative (WaCky) project, two large-scale English corpora have been created.[1] The ukWaC corpus was developed by crawling the `.uk` domain, resulting in nearly 2-billion words then annotated with part-of-speech tags and lemmas (Ferraresi et al., 2008). ukWaC was later extended to include dependency parses extracted using the MaltParser (Nivre et al., 2007) (PukWaC). PukWaC thus represents a large amount of British English text, less formally edited than newswire. The WaCkypedia_EN corpus contains roughly 800-million tokens from a 2009 capture of English Wikipedia, with the same annotations as PukWaC.

Here we relied on the Stanford typed dependencies, rather than the Malt parser, owing to their relative dominance in recent work in distributional semantics and information extraction. In comparison to previous annotated corpora, Annotated Gigaword is a larger resource, based on formally edited ma-

[1] `http://wacky.sslmit.unibo.it/doku.php?id=corpora`

Proc. of the Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX), pages 95–100,
NAACL-HLT, Montréal, Canada, June 7-8, 2012.

terial, that has additional levels of annotation, and reflects the current state of the art in text processing.

In particular, our collection provides the following for English Gigaword v.5 (referred to as *Gigaword* below):

1. tokenized and segmented sentences,
2. Treebank-style constituent parse trees,
3. syntactic dependency trees,
4. named entities, and
5. in-document coreference chains.

The following provides motivation for such a resource, the tools employed, a description of the programmatic interface provided alongside the data, and examples of ongoing work already enabled by this resource.

2 Motivation

Our community has long had a strong dependence on syntactically annotated corpora, going back at least as far as the Brown corpus (Francis and Kučera, 1964 1971 1979). As manual annotation of syntactic structure is expensive at any large scale, researchers have regularly shifted their reliance to automatically parsed corpora when concerned with statistics of co-occurrence.

For example, Church and Hanks (1990) pioneered the use of Pointwise Mutual Information (PMI) in the field, with results provided over syntactic derivations on a 44-million-word corpus of newswire, showing correlations such as the verb *drink/V* associating with direct objects *martinis*, *cup_water*, *champagne*, *beverage*, *cup_coffee*, and so on. This was followed by a large number of related efforts, such as that by Lin and Pantel (2001): Discovery of Inference Rules from Text (DIRT), aimed at building a collection of paths sharing distributionally similar nominal anchors, over syntactic dependency structures automatically derived from newswire text.

While these efforts are popularly known and constitute established methodological baselines within knowledge acquisition and computational semantics, the underlying annotated corpora are not public resources. As such, direct comparison to their methods are difficult or impossible.

Further examples of popularly known results that are difficult to reproduce include the large-scale information extraction results surrounding TextRunner (Yates et al., 2007), or the script induction efforts first described by Chambers and Jurafsky (2008). In the latter, coreference chains were required in addition to syntactic parsing: a further computationally expensive requirement.

Often researchers will provide full resultant derived resources, such as the DIRT rules or narrative chains (Chambers and Jurafsky, 2010). While this is to be encouraged (as opposed to merely allowing limited web-based access), there are likely a number of researchers that would prefer to tune, adapt, and modify large-scale extraction algorithms, if only they had ready access to the preprocessed collections that led to such resources. This is especially the case now, as interest in Vector Space Models (VSMs) for semantics gain increased attention within Cognitive (Mitchell and Lapata, 2010) and Computer (Turney and Pantel, 2010) Science: such models are often reliant on co-occurrence counts derived over large numbers of syntactically analyzed sentences.

3 Annotations

Gigaword was annotated in three steps: (1) preprocess the data and identify which sentences were to be annotated, (2) derive syntactic parses, and (3) post-process the parsed output to derive syntactic dependencies, named entities, and coreference chains. The second step, parsing, took the majority of our efforts: 10.5 days, using 16 GB of memory and 8 cores per Gigaword file. Using six machines, each with 48 cores and 128 GB of memory, we parsed roughly 700-thousand lines per hour.

3.1 Preprocessing

Gigaword has an SGML-style markup which does not differentiate between different types of body text. For example, list items are not distinguished from complete sentences. Therefore, we coarsely identified all non-sentential lines (list items) by lines with more than one character preceding the first non-space character, after inspection of several randomly sampled documents.

The remaining lines from the `<HEADLINE>` and

<TEXT> fields were segmented into sentences using the open-source tool Splitta, which reported the lowest error rate for English sentence segmentation (Gillick, 2009). Sentences were tokenized using a Penn-Treebank tokenizer (from the Stanford CoreNLP toolkit[2]). We skipped all sentences with more than 100 tokens because we observed that these sentences were often the result of sentence segmentation failure or concatenated list items. In total, we parsed 183,119,464 sentences from the collection. Our release includes information about which sentences were omitted. In an initial estimate of one file containing 548,409 sentences, we dropped 1,197 sentences due to length constraints, which is less than one percent of the total sentences.

3.2 Parsing

We have Penn-Treebank-style parses for the 183-million sentences described above, using the state-of-the-art co-trained English parser described in Huang et al. (2010). After consulting the authors, we used the self-trained (using product model) sixth-round grammar (ST-Prod grammar), because it had high accuracy[3] without the exceptional computational burden of a full product of grammars (which was expected to provide only slight improvement, but at significant computational cost).

3.3 Post-Syntactic Processing

We modified the Stanford CoreNLP pipeline to make use of the parse trees from the previous step, in order to then extract dependency structures, named entities, and coreference chains.[4]

Three types of dependency structures were generated and stored: basic typed dependencies, collapsed dependencies, and collapsed dependencies with conjunction dependencies propagated. See de Marneffe and Manning (2008) for details.

We used the best performing coreference-resolution system (Lee et al., 2011) to extract coreference chains over the approximately 180-million sentences in the <TEXT> of each document.

[2] http://nlp.stanford.edu/software/corenlp.shtml

[3] Avg. F score = 91.4 on WSJ sec 22

[4] The Stanford CoreNLP pipeline assumes all aspects of processing are performed with its own tools; the modifications were required to replace the parsing component with an external tool.

3.4 Storage

The data is stored in a form similar to the original Gigaword formatting along with XML annotations containing our additional markup. There is one file corresponding to each file distributed with the Gigaword corpus. The total uncompressed size of the collection is 400 GB, while the original Gigaword is about 26 GB, uncompressed.

4 Programmatic Access

We provide tools for reading the annotated data, including a Java API which provides convenient object representations for the contents of the XML files. Where appropriate, we use the original Stanford toolkit objects, such as *TypedDependency* and *WordLemmaTag*.

We also provide a suite of command-line tools, built on the Java API, for writing out each individual type of annotation in a common text annotation format. For example, one can print out only the part-of-speech tags, or only the dependencies for all the documents in an annotated file.

To parse the XML, we use the VTD-XML[5] parsing model (Zhang, 2008) and its open-source implementation for a 64-bit Java Virtual Machine. The VTD-XML parser allows for random access, while maintaining a very small memory footprint by memory mapping the XML file and maintaining an in-memory index based on the Virtual Token Descriptor (VTD), a concise binary encoding of the XML tokens. Building on VTD-XML, we also provide a streaming mode that processes and keeps in memory only one news document at a time.

The efficiency and ease of extensibility of our tool are a byproduct of it being built on the VTD-XML library. As an example, to parse one XML file (470 MB) consisting of 33,108 sentences into an object-oriented representation of the dependency parses and accumulate sufficient statistics about dependency edge counts requires just over 30 seconds using a 64 MB of heap space and a single core of an Intel Xeon 2.66 Ghz CPU.

[5] http://vtd-xml.sourceforge.net

Path	Gloss	Cos
NNS:nsubj:VBD← *dived*→VBD:dobj:NN	X dived Y	1.0000
NNS:nsubj:VBD←*slumped* →VBD:dobj:NN	X slumped Y	0.9883
NNS:nsubj:VBD←*plunged*→VBD:dobj:NN	X plunged Y	0.9831
NNS:nsubj:VBD←*gained*→VBD:dobj:NN	X gained Y	0.9831
NNS:nsubj:VBD←*soared*→VBD:dobj:NN	X soared Y	0.9820
NNS:nsubj:VBD←*leapt*→VBD:dobj:NN	X leapt Y	0.9700
NNS:nsubj:VBD←*eased*→VBD:dobj:NN	X eased Y	0.9700
NNS:pobj:IN←of←IN:prep:NN←index←NN:nsubj:VBD←*rose*→VBD:dobj:NN	X's index rose Y	0.9685
NNS:nsubj:VBD←*sank*→VBD:dobj:NN	X sank Y	0.9685
NNS:pobj:IN←of←IN:prep:NN←index←NN:nsubj:VBD←*fell*→VBD:dobj:NN	X's index fell Y	0.9621

Table 1: Relations most similar to "X dived Y" as found in Annotated Gigaword using approximate search.

Path	Gloss	Cos
NN:nsubj:VBD←*gained*→VBD:dobj:NNS	X gained Y	1.0000
NN:nsubj:VBD←*climbed*→VBD:dobj:NNS	X climbed Y	0.9883
NN:nsubj:VBD←*won*→VBD:dobj:NNS	X won Y	0.9808
NN:nsubj:VBD←*rose*→VBD:dobj:NNS	X rose Y	0.9783
NN:nsubj:VBD←*dropped*→VBD:dobj:NNS	X dropped Y	0.9743
NN:nsubj:VBD←*edged*→VBD:dobj:NNS	X edged Y	0.9700

Table 2: Relations most similar to "X gained Y" as found in Annotated Gigaword using approximate search.

5 Example Applications

The following gives two examples of work this resource and interface have already enabled.[6]

5.1 Shallow Semantic Parsing

Ongoing work uses this resource to automatically extract relations, in the spirit of Lin and Pantel (2001) (DIRT) and Poon and Domingos (2009) (USP). First, DIRT-like dependency paths between nominal anchors are extracted and then, using these observed nominal arguments to construct feature vectors, similar paths are discovered based on an approximate nearest-neighbor scheme as employed by Ravichandran et al. (2005). For example, the most similar phrases to "X dived/gained Y" found using this method are shown in Tables 1 and 2 (e.g. *the Nasdaq dived 3.5 percent*). Deriving examples such as these required relatively minor amounts of effort, but only once a large annotated resource and supporting tools became available.

[6]Both applications additionally rely on the Jerboa toolkit (Van Durme, 2012), in order to handle the large scale of features and instances extractable from Annotated Gigaword.

5.2 Enabling Meaning-preserving Rewriting

In a related project, Annotated Gigaword enabled Ganitkevitch et al. (2012) to perform large-scale extraction of rich distributional signatures for English phrases. They compiled the data into a flat corpus containing the constituency parse, lemmatization, and basic dependencies for each sentence. For each phrase occurring in the sentence, contextual features were extracted, including:

- Lexical, lemma, and part-of-speech n-gram features, drawn from an m-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, lemma, and part of speech. If the phrase was syntactically well-formed, lexical, lemma, and part-of-speech features for its head were also included.
- Syntactically informed features for constituents governing the phrase, as well as for CCG-style slashed constituent labels for the phrase, into individual features by governing constituent and left- or right-missing constituent.

These features extracted from Annotated Gigaword were successfully used to score paraphrase similarity in a text-to-text generation system. Due to its much more diverse feature set, the resulting collection of 12-million rich feature vectors yielded significantly better output (as judged by humans) than a vastly larger collection of 200-million phrases derived from a web-scale n-gram corpus.

6 Conclusion

As interest in methods requiring large-scale data continues to grow, it becomes ever more important that standard reference collections of preprocessed collections be made available. Annotated Gigaword represents an order of magnitude increase over syntactically parsed corpora currently available via the LDC. Further, it includes Stanford syntactic dependencies, a shallow semantic formalism gaining rapid community acceptance, as well as named-entity tagging and coreference chains. Throughout we have relied on state-of-the-art tools, providing researchers a level playing field to experiment with and compare methods for knowledge acquisition and distributional semantics.

Acknowledgments

The authors wish to thank Juri Ganitkevich, Xuchen Yao, Chris Callison-Burch, Benjamin Shayne and Scott Roberts for their feedback and assistance. This work was partly supported by the National Science Foundation under Grant Nos. DGE-0707427 and DGE-1232825, and by the Johns Hopkins University Human Language Technology Center of Excellence.

References

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics: Human Language Technologies (ACL08: HLT)*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2010. A database of narrative schemas. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. 2000. *BLLIP 1987-89 WSJ Corpus Release 1*. Linguistic Data Consortium.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. *http://nlp.stanford.edu/software/dependencies_manual.pdf*.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

W. Nelson Francis and Henry Kučera. 1964, 1971, 1979. A Standard Corpus of Present-Day Edited American English, for use with Digital Computers (Brown).

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2012. Monolingual distributional similarity for text-to-text generation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*.

Dan Gillick. 2009. Sentence boundary detection and the problem with the US. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244, Boulder, Colorado. Association for Computational Linguistics.

Zhongqiang Huang, Mary Harper, and Slav Petrov. 2010. Self-training with products of latent variable grammars. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 12–22, Cambridge, MA. Association for Computational Linguistics.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, Oregon, USA. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. DIRT: Discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 323–328, New York, NY, USA. ACM.

David McClosky, Eugene Charniak, and Mark Johnson. 2008. *BLLIP North American News Text, Complete*. Linguistic Data Consortium.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Stetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. *English Gigaword Fifth Edition*. Linguistic Data Consortium.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Singapore. Association for Computational Linguistics.

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 the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 622–629, Ann Arbor, Michigan. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research (JAIR)*, 37(1):141–188.

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

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. 2007. TextRunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA. Association for Computational Linguistics.

Jimmy Zhang. 2008. VTD-XML: XML processing for the future (Part I). http://www.codeproject.com/Articles/23516/VTD-XML-XML-Processing-for-the-Future-Part-I.