Cohere: A Toolkit for Local Coherence

Karin Sim Smith§, Wilker Aziz†, Lucia Specia§
§Department of Computer Science, University of Sheffield, UK
†Institute for Logic, Language and Computation
University of Amsterdam, The Netherlands
{kmsimsmith1,l.specia}@sheffield.ac.uk , w.aziz@uva.nl

Abstract

We describe cohere, our coherence toolkit which incorporates various complementary models for capturing and measuring different aspects of text coherence. In addition to the traditional entity grid model (Lapata, 2005) and graph-based metric (Guinaudeau and Strube, 2013), we provide an implementation of a state-of-the-art syntax-based model (Louis and Nenkova, 2012), as well as an adaptation of this model which shows significant performance improvements in our experiments. We benchmark these models using the standard setting for text coherence: original documents and versions of the document with sentences in shuffled order.

Keywords: Discourse, Coherence Models, Local Coherence

1. Introduction

Coherence can be defined as what holds a text together and makes it logical and understandable for the reader. This is achieved by various means, some of which are more amenable to computational models than others.

The only freely available coherence software we are aware of is the Brown Coherence Toolkit.[1] It provides a range of entity-grid-based coherence models, capturing information related to entities that are common to adjacent sentences. Our toolkit offers additional, more varied models which include syntax, and in the case of the graph-based metric, capture links throughout the text.

In what follows, we briefly describe the models in our toolkit (Section 2.), provide information on how to use the toolkit (Section 3.), and show our experiments with it (Section 4.).

2. Models in cohere

We briefly describe our own implementation of the most popular model for text coherence, the entity grid model, in Section 2.1.. Then in Section 2.2. we describe our entity graph implementation, which captures different aspects of lexical coherence from an entity grid, tracking connections between non-adjacent entities in the text. This model covers aspects of coherence that reflect the centrality and topic of the discourse. Following that we present our implementation of the syntax-based coherence model of Louis and Nenkova (2012), which captures syntactic patterns between adjacent sentences (Section 2.3.). Finally, in Section 2.4. we present our own extension of this model, a fully generative model incorporating IBM Model 1 (Brown et al., 1993) to model alignments over syntactic items in adjacent sentences.

2.1. Entity Grid Model

As detailed in (Barzilay and Lapata, 2008; Elsner et al., 2007), the entity-based approach derives from the assumption that entities in a coherent text are distributed in a certain manner, as posed by various discourse theories, in particular Centering Theory (Grosz et al., 1995). This theory holds that coherent texts are characterised by salient entities in strong grammatical roles, such as subject or object. Entity grids are constructed by identifying the discourse entities in the documents under consideration, and constructing a 2D grid for each document, whereby each column corresponds to the entity, i.e. noun, being tracked, and each row represents a particular sentence in the document.

An **entity transition** is defined as a consecutive occurrence of an entity with a given syntactic role, namely, subject (S), object (O), or other (X). Transitions are observed by examining the grid vertically for each entity. We replicate the generative model of document coherence based on entity transitions introduced by Lapata (2005). Equation 1 shows this formulation, where m is the number of entities, n is the number of sentences in a document D and $r_{s,e}$ is the role taken by entity e in sentence s.

$$p(D) = \frac{1}{m \cdot n} \prod_{e=1}^{m} \prod_{s=1}^{n} p(r_{s,e} | r_{(s-h),e} \dots r_{(s-1),e}) \quad (1)$$

This is because the syntactic patterns which hold for English, do not hold for German, for example (Cheung and Penn, 2010).

2.2. Entity Graph Model

Guinaudeau and Strube (2013) converted a standard entity grid into a bipartite graph which tracks the occurrence of entities throughout the document, including between non-adjacent sentences.

A local coherence score is calculated directly as the average outdegree of a projection, summing the shared edges of entities between two sentences. The general form of the coherence score assigned to a document D in this approach is shown in Equation 2. This is a centrality measure based on the average outdegree across the N sentences represented in a directed document graph. The outdegree of a sentence s_i, denoted $o(s_i)$, is the total weight of all the edges leaving that sentence , a notion of how connected (or how central) it is. This weight is the sum of the contributions of all edges connecting sentence s_i to any other sentence $s_j \in D$.

[1] http://cs.brown.edu/melsner/manual.html

$$s(D) = \frac{1}{N}\sum_{i=1}^{N} o(s_i) \tag{2}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=i+1}^{N} W_{i,j}$$

Guinaudeau and Strube (2013) define 3 types of graph projections: *binary*, *weighted* and *syntactic*.

Binary, or unweighted, projections simply record whether two sentences, s_i and s_j, have any entities in common. *Weighted* projections take the number of shared entities into account, rating the projections higher for more shared entities. A *syntactic* projection includes syntax information, where this information is used to weight the importance of the link: an entity in role of subject (S) is weighted as a 3, an entity in role of object (O), as a 2, and other (X), as a 1. Our implementation incorporates all three types of projection. Again, if a dependency parser is not available, for a certain language, the projections can only be binary or weighted. And for some languages it does not make sense to use syntactic weights, as they do not reflect the role of the entity in the same way as for English. In particular, it has been proven that the same patterns of syntactic constructions do not hold for eg. German, where topological fields are more relevant (Cheung and Penn, 2010).

2.3. Louis and Nenkova's Syntax Model

Louis and Nenkova (2012) propose both a local and a global coherence model based on syntactic patterns. Our implementation focuses on their local coherence model. It follows the hypothesis that in a coherent text consecutive sentences will exhibit syntactic regularities. Moreover, that these regularities can be captured in terms of cooccurrence of syntactic items.

The units of syntax examined can be context-free grammar productions (e.g. S → NP VP) or d-sequences (a sequence of sibling constituents at depth d starting from the root, possibly annotated with the left-most child node they dominate, e.g. NP_{NN} VP_{VB}). The model conditions each sentence on the immediately preceding sentence, both seen as sequences of syntactic patterns. Each sentence is assumed to be generated one pattern at a time and patterns are assumed independent of each other. The parameters of the model are "unigram" and "bigram" patterns over a vocabulary of syntactic items (i.e. productions or d-sequences). The coherence of a document under the model is given by Equation 3, where (u_1^m, v_1^n) represents adjacent sentences, and $c(\cdot)$ is a function that counts how often a pattern (or a pair of patterns) was observed in the training data.

$$p(D) = \prod_{(u_1^m, v_1^n)\in D} \prod_{j=1}^{n} \frac{1}{m}\sum_{i=1}^{m} \frac{c(u_i, v_j) + \alpha}{c(u_i) + \alpha|V|} \tag{3}$$

To account for unseen syntactic patterns at test time, the model is smoothed by a constant α.

2.4. Syntax Model with IBM 1 Alignments

A way to understand the model by Louis and Nenkova (2012) is to see it as an alignment model between syntactic items. However, that model does not have any latent variables, which is possible under the assumption that all available alignment configurations have been directly observed in the training data. It is worth highlighting that in reality the training data is incomplete, in the sense that it lacks alignment information. Thus, we introduce alignments between syntactic patterns in adjacent sentences as a latent variable. Our model is in fact an instance of IBM model 1 (Brown et al., 1993), where the current sentence is generated by the preceding one, one pattern at a time, with a uniform prior over alignment configurations. The latent alignment variable allows us to model the fact that some patterns are more likely to trigger certain subsequent patterns.

In IBM Model 1, a latent alignment function $a : j \mapsto i$, maps patterns in v_1^n (the current sentence) to patterns in u_0^m (the preceding sentence), where u_0 is a special NULL symbol which models insertion of patterns. The score of a document is thus:

$P(D) = \prod_{(u_1^m, v_1^n)\in D} p(v_1 \dots v_n, a_1 \dots a_n | u_0 \dots u_m)$.

As mentioned above, here n is the current sentence and m the preceding sentence. As the alignment is hidden, we marginalise over all possible configurations, which is tractable due to the independence assumption (that items align independently of each other). Equation 4 shows this tractable marginalisation.

$$p(D) = \prod_{(u_1^m, v_1^n)\in D} \prod_{j=1}^{n} \sum_{i=0}^{m} p(v_j|u_i) \tag{4}$$

We resort to Expectation Maximisation (EM) to estimate the parameters in Equation 4 (Brown et al., 1993). Due to the convexity of IBM Model 1, EM is guaranteed to converge to a global optimum. Moreover, as we observe more data, this model converges to better parameters. A similar solution was proposed in a different context by (Soricut and Marcu, 2006) in their work on word co-occurrences.

To avoid assigning 0 probability to documents containing unseen patterns, we modify the original training procedure to treat all the singletons as belonging to an unknown category (UNK), thus reserving probability mass for future unseen items.

3. Running the Toolkit

Input The input data for the toolkit is raw text, with markup for document breaks. The syntax models take ptb marked up files, which can be derived from code included in the toolkit.

Models available

- Entity Grid model, our own generative grid model in line with Lapata (2005).
- Entity Graph metric, implementation of Guinaudeau and Strube (2013).
- Syntax-based model, based on Louis and Nenkova (2012).
- Our syntax-based model with IBM Model 1 alignments (over syntactic items in adjacent sentences).

Parameters

- depth: this parameter can be used to vary the depth of the syntactic trees (default is 3). This applies to the syntax-based model and the syntax-based model with IBM 1 alignments.
- salience: this parameter is used by our entity grid model to define how salient the entity is, e.g. 2 would mean that only entities occurring more than twice are considered.
- projection: this parameter is used by our graph metric, to determine the projection, i.e. *binary*, *weighted* and *syntactic*.

Output The above models output a score per document. These can be used, for example, to automatically compare or rank alternative versions of a documents, or more generally for comparisons across documents.

4. Experiments and Results

We conducted a series of experiments to test our toolkit and to test our hypothesis that patterns of syntactic items between adjacent sentences can be better modelled through a latent alignment model. In other words, a second aim was to check whether our IBM Model 1 formulation for the syntax model outperforms the original syntax model.

We follow the standard approach of comparing a human-authored coherent text with a shuffled version of it, where the order of sentences is randomly altered. This corresponds to the binary discrimination task described in the Brown Coherence Toolkit. For the experiments we used a POS tagger[2] to identify the nouns and subsequently a parser[3] to establish the grammatical role of each of these nouns.

4.1. Data

To estimate the parameters of the entity-grid and syntax-based models (e.g. distribution over entity role transitions and syntactic patterns), we use the most recent portion of English LDC Gigaword corpus,[4] part 12/2010, which encompasses 41,564 documents and 774,965 sentences, excluding two sections deemed inadequate quality. To test our models, we take a standard corpus widely used for coherence evaluation, the Earthquakes and Accidents corpus, with the given permutations[5]

4.2. Results

Tables 1 (Accidents) and 2 (Earthquakes) show the performance of our models according to different evaluation methods (scores are percentages). Our evaluation metrics are defined as follows:

ref$_{1^*}$ how often the reference (i.e., original, human-authored documents) is ranked strictly higher than any of their shuffled counterparts. (In the case of this dataset, there are 20 shuffled versions[6]): $\frac{1}{|D|}\sum_d \text{first}_m(d_r) \times \text{solo}_m(d_r)$

ref$_{\geq}$ how often a model ranks the reference no worse than any of the shuffled counterparts: $\frac{1}{|D||S|}\sum_d \sum_s \text{win}_m(d_r, d_s) + \text{tie}_m(d_r, d_s)$

Our results are shown in Tables 1 and 2.

In our experiments with the syntax models, we derived the syntactic items in the form of the d-sequence, defined as the leaves of the parse tree at a given depth (in our experiments of depth 2, 3, 4), and annotated with the left-most leaf. We display results for both IBM1 and LOUIS at varying levels of depth in Tables 1 and 2 with suffix $-d2$,$-d3$,$-d4$. We display the best results for the syntax models obtained over these depths.

It is clear that our IBM Model 1 formulation for the syntax model outperforms the original syntax model. It also outperforms our implementation of the entity grid, and the entity graph metric for one of the languages.

We used the syntactic projection for our Graph metric. It performs well, despite the fact that it requires no training, unlike the other models. We believe this is because it manages to capture an essence of the web of entity connections which encompass the entire document. Interestingly it does not do so well on the Earthquakes corpus, which other models generally do better on. We surmise this is due to the fact that the other models can capture the particular sequence of the very structured report that is represented in this corpus; in particular this suits our IBM Model 1 formulation, as it tracks syntactic patterns in adjacent sentences in a more stringent manner. Whereas the directed graph measures all the connected entities, but with the repetition of entities throughout the short text perhaps struggles to differentiate the order more. Often the main entities of the document in this corpus are in the first line. The graph will then score the edges from them to any subsequent sentences.

Results for previous grid experiments (Barzilay and Lapata, 2008) were obtained from supervised training, whereby the parameters are trained on this same Earthquakes and Accidents corpus, then tested on a heldout section of the same dataset. We adopted a more automated approach, training on more general data, with a view to being applied more widely. This does, however, affect the results, particularly given the nature of the Earthquakes and Accidents corpus. It consists of short articles (averaging 10.4 and 11.5 sentences in length, respectively) (Barzilay and Lapata, 2008), with many short sentences and with a very consistent structure. The scores we report are therefore not as high as those reported in previous work, because we have not trained our models to the fixed format of that corpus.

It is worth pointing out that both our reimplementation of the original syntax model and our IBM1 model were trained on the aforementioned LDC data. This again accounts for the slightly lower figures by comparison with the original syntax model.

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

[3] http://nlp.stanford.edu/software/lex-parser.shtml

[4] https://catalog.ldc.upenn.edu/LDC2003T05

[5] http://people.csail.mit.edu/regina/coherence/CLsubmission/

[6] http://people.csail.mit.edu/regina/coherence/CLsubmission/

Model	ref$_{1^*}$	ref$_{\geq}$
Graph	86.51	86.51
IBM1-d3	72.61	72.61
IBM1-d2	67.32	67.37
Grid	50.25	50.25
Louis-d4	46.58	55.89
Louis-d2	38.82	57.15

Table 1: Model comparisons for shuffling experiment on accident corpora, ref$_{1^*}$ is "accuracy" used in previous work with this corpus.

Model	ref$_{1^*}$	ref$_{\geq}$
IBM1-d2	80.88	80.88
IBM1-d3	77.10	77.10
Grid	66.21	66.21
Graph	60.53	60.58
Louis-d2	57.62	71.73
Louis-d3	57.00	67.69

Table 2: Model comparisons for shuffling experiment on earthquake corpora, ref$_{1^*}$ is "accuracy" used in previous work with this corpus.

5. Conclusions

We introduced a coherence toolkit : `cohere`. It offers additional and extended models to measure local coherence, as compared to toolkits currently available. `cohere` incorporates models that consider syntactic patterns within a discourse and assess coherence on that basis. The toolkit can be easily extended to add new models, as well as to test various combinations of models. It is also versatile in the tasks it can be used for, as the input to the model is simply raw text.

The tool is distributed under a permissive BSD license and can be downloaded from https://github.com/karins/CoherenceFramework.

6. Bibliographical References

Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Comput. Linguist.*, 34(1):1–34, March.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Cheung, J. C. K. and Penn, G. (2010). Entity-based local coherence modelling using topological fields. In *ACL*, pages 186–195.

Elsner, M., Austerweil, J., and Charniak, E. (2007). A unified local and global model for discourse coherence. In *Proceedings of HLT-NAACL*, pages 436–443.

Grosz, B. J., Weinstein, S., and Joshi, A. K. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21:203–225.

Guinaudeau, C. and Strube, M. (2013). Graph-based local coherence modeling. In *Proceedings of ACL*, pages 93–103.

Lapata, M. (2005). Automatic evaluation of text coherence: models and representations. In *Proceedings of IJCAI*, pages 1085–1090.

Louis, A. and Nenkova, A. (2012). A coherence model based on syntactic patterns. In *Proceedings of EMNLP-CoNLL*, pages 1157–1168, Jeju Island, Korea.

Soricut, R. and Marcu, D. (2006). Discourse generation using utility-trained coherence models. In *Proceedings of the COLING/ACL*, pages 803–810, Sydney, Australia.