What are Ontologies Good For? Evaluating Terminological Ontologies in the Framework of Text Graph Classification

— Extended Abstract —

Alexander Mehler[1] and Angelika Storrer[2]

[1] Bielefeld University
Alexander.Mehler@uni-bielefeld.de
[2] Dortmund University
Angelika.Storrer@uni-dortmund.de

Abstract. This paper develops a graph-theoretical model of text representation based on lexical chaining. Other than present approaches to chaining, this model reflects the logical document structure of texts as well as semantic relations of their lexical constituents in order to compute text similarity values. By varying the terminological ontology used to induce such relations, a door is opened to systematically evaluate their contribution to text classification. This is exemplified by example of GermaNet and the Wikipedia.

1 Introduction

With the accessibility of large corpora of natural language texts, the corpus-based exploration of collocations became a standard task in computational linguistics [14]. These collocations have been further explored in order to induce sense relations (e.g. synonymy or hyponymy) of lexical units [10]. This, in principle, opened the door to partly operationalize the notion of *textual cohesion*. Generally speaking, cohesion is modeled as a system of linguistic resources that languages use for linking sentences and other text spans to one another. According to Halliday & Hasan [9], its lexical equivalent, i.e. *lexical cohesion*, results from reiteration and collocation, respectively:

- *Reiteration* means the co-occurrence of lexical items which instantiate the same or different, but sense related words, that is, words which enter into some sense relation.
- *Collocation* is the tendency of words to co-occur, for example, because of semantic associations or lexical solidarities.

The contribution of reiteration and collocation to lexical cohesion is that tokens of a text t which are iterations of each other or instantiate words with similar patterns of collocation contribute to the cohesion of t if they occur in the same or adjacent sentences [9].

In computational linguistics, reiteration is modeled with the help of lexical reference systems like Roget's Thesaurus, WordNet [4, 18, 25] or GermaNet [15]. On the other hand, collocations are explored by means of (e.g. similarity) functions of lexical co-occurrences [14, 21]. There also exist models which integrate both approaches in order to predict lexical cohesion in a unified model of reiteration and collocation [24].

A candidate for bridging the notion of lexical cohesion on the one hand and that of reiteration and collocation on the other hand is the concept of a *lexical chain*. A lexical chain of a text t is a sequence of semantically related tokens of t [18]. *Lexical chaining* is the task of exploring such chains, that is, tracking semantically related tokens in a text. Applications of chaining range from *text segmentation* [5, 11, 16, 18], *summarization* [1], *topic clustering* [19], *topic tracking* [24], *generating alternative text views* [25] to the *detection of malapropisms* [12]. Green [6, 7, 8] exploits lexical chains in order to generate intra- and intertextual hyperlinks. More specifically, he focuses on pairwise paragraph-to-paragraph and text-to-text links. The concept of pairwise linking relates to the greedy nature of approaches which try to link items as early as possible irrespective of structure formation [5]. Cf. Mehler [17] who develops a chaining algorithm in order to induce intertextual links which partly overcome this greedy nature.

In this paper, we utilize lexical chaining as a text representation model in the framework of a classification task. That is, we propose a graph-theoretical representation of texts based on their lexical chains as a resource of text similarity measuring which, in turn, is utilized in text classification. Our starting point is Green's lexical chaining model. Essentially, Green describes a four-stage model of text representation: Firstly, an input text t is represented as a set of lexical chains each of which links a set of tokens of t as instances of content words which are interrelated according to some WordNet relation (for the details of this step cf. [7]). Secondly, t is represented as a sequence of so called *chain density vectors* each of which represents a separate paragraph of t. An element of the chain density vector $\boldsymbol{v}$ of a paragraph p of t estimates the density of the corresponding lexical chain c in p, that is, the fraction of content words of p that also belong to c. Thirdly, the chain density vectors are input to inferring inter-paragraph links based on some similarity measure (e.g. the cosine measure). This leads to a graph model of a single text as illustrated in Figure (1). Fourthly, Green represents t as a so called *synset weight vector* (swv) $\boldsymbol{w}$. An element of $\boldsymbol{w}$ stands for a corresponding synset of WordNet; its value weighs the number of occurrences of this synset in the lexical chains of t. The similarities of the swvs $\boldsymbol{w}_i, \boldsymbol{w}_j$ of different texts t_i, t_j are, finally, explored in order to infer intertextual links of these texts. This similarity measure is insensitive to text structure. The reason is that swvs are derived from lexical chains whose order and extent do not affect their values (apart from their size). Actually, swvs can be computed by considering chaining only to the extent that it determines the tokens which count as relevant content words. This also holds for paragraph-to-paragraph linkage based on chain density vectors which do not reflect the order and extent of the underlying chains. Although this observation is unproblematic from the point

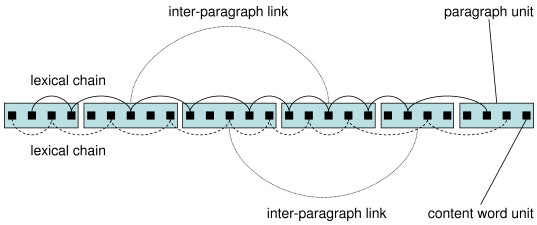

Fig. 1. Illustration of the graph-theoretical model of text representation based on lexical chaining according to Green [6, 7].

of view of the task performed by Green's algorithm, i.e. hypertext generation, we propose a text representation model in which the structure of the chains is preserved in order to get insights into the information value of the terminological ontologies and word nets used to generate them.

An alternative graph-theoretical model of text representation is developed by Schenker et al. [20] who propose a series of models which consist of labeled graphs whose vertices denote lexical items and whose edges denote co-occurrences of these items within the corresponding input text. In some of these models, edges are labeled by the type (e.g. title, text [body] etc.) of the constituent of document structure in which the respective co-occurrence is observed. Schenker et al. also consider weighted edges based on the number of co-occurrences. In any case, the graphs inferred in this setting only rely on lexical items as found in the input text without taking their semantic relations into account (not to mention relations to words which do not occur in the input text). Consequently, this approach suffers from the synonymy and polysemy problem as described by Green [7]. Nevertheless, the central merit of this model is the graph distance measure and clustering algorithm it provides. Schenker et al. prove that any pair of instances of their graph models allow to compute their maximum common subgraph far away from the NP-completeness of the general case of this algorithm – actually, in the present case, this computation is of order $O(|V|^2)$, $V = \max(|V_1|, |V_2|)$, where V_1, V_2 are the vertex sets of the input graphs.

It is this apparatus which we reutilize in the present paper, but by developing a graph-theoretical model of text representation which – *other than the model of Green* – is sensitive to text structure and which – *other than the model of Schenker et al.* – is sensitive to lexical semantic relations. A reference point of this model is the choice of the terminological ontology [23] used to infer graph representations of texts. By systematically varying this parameter we get access to evaluating the varying contributions of different terminological ontologies and, thus, to rating their information value in text mining *by example of the task of text classification.*

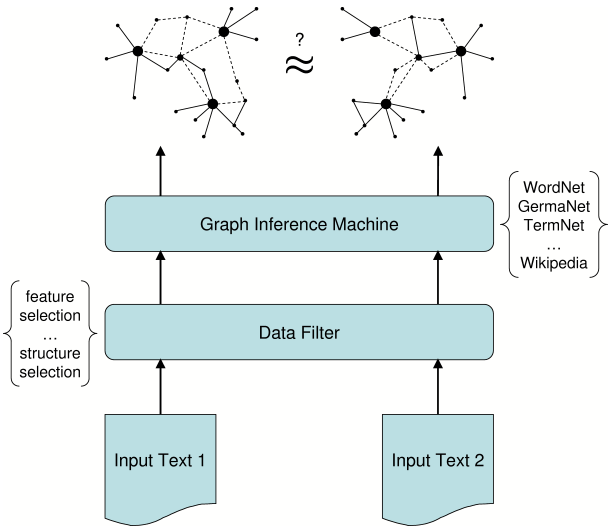

Fig. 2. The standard scenario of graph similarity measuring based on inferring graph representations of two input texts subject to different terminological ontologies and word nets.

2 The Graph Inference Machine

The basic idea of our approach is to build a graph-theoretical model of text representation which combines the text *syntactic* orientation of Schenker et al.'s approach with the lexical *semantic* orientation of Green's approach. More specifically, we propose a *Graph Inference Machine* (GIM) in order to infer labeled multigraphs for input texts by taking three sorts of edges into account:

1. *Reiteration links* are induced according to the algorithm of Green. They are labeled as r.
2. *Collocation links* are induced according to some method of collocation analysis (e.g. *latent semantic analysis* [14]), where tokens are connected by a collocation link if the similarity of their collocation patterns exceeds a certain threshold. Edges of this kind are labeled as c.
3. Finally, *structure links* between tokens codify dependency or constituency relations as, for example, '*occurring in the same unit*' (e.g. abstract, paragraph or section) of the logical document structure of the input text. Edges of this kind are labeled by an identifier of the corresponding unit (e.g. abstract, first paragraph etc.).

These are alternative resources for inferring edges between vertices which denote tokens of a given input text. Thus, edge formation can take reiteration *or* collocation *or* structure links into account. As the inference of reiteration links is, in turn, dependent on the underlying terminological ontology (e.g. Roget's thesaurus vs. WordNet vs. the Wikipedia and its category system) and since the

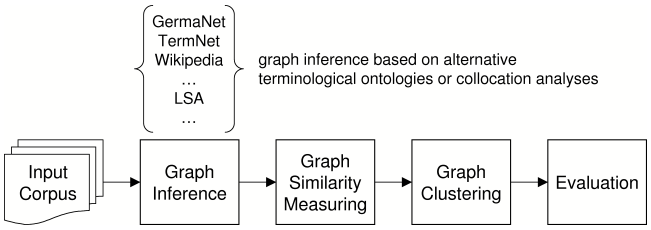

Fig. 3. The standard evaluation scenario.

induction of collocation links depends on the choice of the method of collocation analysis, the GIM is associated with a corresponding parameter space. This is illustrated in Figure (2) in which data filtering precedes graph inference. The task of the GIM is to infer alternative text representations subject to the choice of values of its parameters. Note that this choice may combine different terminological ontologies (e.g. GermaNet in conjunction with TermNet [2]) as input to the inference machine. It outputs graphs for input texts which can, then, be asked for their similarity. These graph similarities open the door to evaluating the information value of terminological ontologies subject to their positive or negative effect on graph classification as explained in the following section. In the present paper, we concentrate on the contribution of GermaNet and Wikipedia.

3 The Evaluation Setting

We perform an automated, standardized evaluation procedure which does not only guarantee repeatability, but also comparability with future developments in the present field. For this purpose, we extend the set of text representation models of text classification and categorization [22] by our GIM-induced models and perform a standard classification experiment [3] in an unsupervised fashion. This scenario is outlined in Figure (3): Starting from a corpus $C = \{x_1, \ldots x_n\}$ of documents (e.g. natural language texts, web pages, websites etc.), a set of classes $\mathbb{C} = \{c_1, \ldots, c_k\}$ and a partitioning $\mathbb{L}$ of C into disjunct subsets $\cup_{i=1}^{k} L_i = C$, such that L_j is the set of textual manifestations of the class $c_j \in \mathbb{C}$, $j \in \{1, \ldots, k\}$, a partitioning $\mathbb{P}$ of C is learned and evaluated in four steps (cf. Figure 3):

1. *Graph inference* is performed by means of the graph inference machine of Section (2). According to the varying choice of values of its parameter space, this results in alternative graph-theoretical text representations to be evaluated subsequently.
2. *Graph similarity measuring* is done by means of *maximum common subgraph*-based methods [20]. As different similarity measures can be applied at this point, they extend the parameter space of our approach.
3. *Graph clustering* operates on the graph similarity matrix computed in the preceding step. It applies standard methods of hierarchical and partitive

cluster analysis which further extend our parameter space. This step outputs a partition $\mathbb{P}$ to be compared with the "golden standard" $\mathbb{L}$.
4. Finally *graph cluster evaluation* is performed by means of the F-measure by comparing $\mathbb{P}$ with $\mathbb{L}$ [13].

As input we process a corpus of newspaper articles whose partition $\mathbb{L}$ is due to the pre-established mappings of these articles to topics, rubrics and genre categories as found within these texts. In order to evaluate the contribution of terminological ontologies – *as a resource of measuring lexical cohesion due to reiteration* – in comparison to a baseline algorithm – *as a resource of measuring lexical cohesion due to collocation* –, we choose the best performing combination of graph similarity measuring and clustering. This allows to vary, ceteris paribus, the resource ontology as well as the method of collocation analysis within the GIM. Note that in this paper, we concentrate on the contribution of terminological ontologies. Independent from the absolute degree of performance, this evaluation results in a statement about the usability of terminological ontologies. It can be referred to in order to rate future developments of such ontologies at least with respect to their usability in text classification tasks.

Bibliography

[1] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, Spain*. 1997.

[2] M. Beisswenger, A. Storrer, and M. Runte. Modellierung eines Terminologienetzes für das automatische Linking auf der Grundlage von WordNet. *LDV-Forum*, 19(1/2):113–125, 2004.

[3] H. H. Bock. Classification and clustering: Problems for the future. In E. Diday, Y. Lechevallier, M. Schader, P. Bertrand, and B. Burtschy, editors, *New Approaches in Classification and Data Analysis*, pages 3–24. Springer, Berlin, 1994.

[4] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, 1998.

[5] O. Ferret. Using collocations for topic segmentation and link detection. In *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002, Taipei*, pages 260–266. Morgan Kaufmann, 2002.

[6] S. J. Green. Automated link generation: Can we do better than term repetition? *Computer Networks and ISDN Systems*, 30(1-7):75–84, 1998.

[7] S. J. Green. Building hypertext links by computing semantic similarity. *IEEE Transaction on Knowledge and Data Engineering*, 11(5):713–730, 1999.

[8] S. J. Green. Lexical semantics and automatic hypertext construction. *ACM Computing Surveys*, 31(4), 1999.

[9] M. A. K. Halliday and R. Hasan. *Cohesion in English*. Longman, London, 1976.

[10] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics, (COLING '92), August 23-28, 1992, Nantes, France*, pages 539–545, 1992.

[11] M. A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[12] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malaproprisms. In C. Fellbaum, editor, *WordNet – An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts, 1998.

[13] A. Hotho, A. Nürnberger, and G. Paass. A Brief Survey of Text Mining. *LDV-Forum*, 20(1):19–62, 2005.

[14] T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[15] L. Lemnitzer and C. Kunze. Adapting GermaNet for the Web. In *Proceedings of the First Global Wordnet Conference*, pages 174–181, Central Institute of Indian Languages, Mysore, India, 2002.

[16] D. Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, Massachusetts, 2000.

[17] A. Mehler. Lexical chaining as a source of text chaining. In J. Patrick and C. Matthiessen, editors, *Proceedings of the First Computational Systemic Functional Grammar Conference, University of Sydney, Australia, July 16*, 2005.

[18] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

[19] M. Phillips. *Aspects of Text Structure. An Investigation of the Lexical Organisation of Text*. North Holland, Amsterdam, 1985.

[20] A. Schenker, H. Bunke, M. Last, and A. Kandel. *Graph-Theoretic Techniques for Web Content Mining*. World Scientific, New Jersey/London, 2005.

[21] H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[22] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[23] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, Pacific Grove, 2000.

[24] N. Stokes. *Application of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain*. PhD thesis, National University of Ireland, Dublin, April 2004.

[25] E. Teich and P. Fankhauser. Exploring lexical patterns in text: Lexical cohesion analysis with WordNet. In *Interdisciplinary studies on information structure*, pages 129–145. SFB 632, Universität Potsdam, 2005.