Natural Language Engineering **21** (5): 665–698. © Cambridge University Press 2015
doi:10.1017/S1351324915000340

A survey of graphs in natural language processing[*]

VIVI NASTASE[1], RADA MIHALCEA[2]
and DRAGOMIR R. RADEV[2]

[1]*Human Language Technologies – Natural Language Processing, Fondazione Bruno Kessler, Trento, Italy*
email: nastase@fbk.eu
[2]*Department of Electrical Engineering and Computer Science and School of Information, University of Michigan, USA*
email: mihalcea@umich.edu, radev@umich.edu

(*Received 24 August 2015; revised 10 September 2015*)

Abstract

Graphs are a powerful representation formalism that can be applied to a variety of aspects related to language processing. We provide an overview of how Natural Language Processing problems have been projected into the graph framework, focusing in particular on graph construction – a crucial step in modeling the data to emphasize the phenomena targeted.

1 Introduction

Graphs are ubiquitous in Natural Language Processing (NLP). They are relatively obvious when imagining words in a lexical resource or concepts in a knowledge network, or even words within a sentence that are connected to each other through what is formalized as syntactic relations. They are less obvious, however still there, when thinking about correcting typos, sentiment analysis, machine translation, figuring out the structure of a document or language generation.

Graphs are a powerful representation formalism. In language, this is probably most apparent in graph-based representations of words' meanings through their relations with other words (Quillian 1968), which has resulted in WordNet (Fellbaum 1998) – a semantic network that after more than 20 years is still heavily used for a variety of tasks (word sense disambiguation, semantic similarity, question answering, and others). Interestingly, some tasks are concerned with updating or expanding it, proof of the usefulness of this representation for capturing lexical semantics, or connecting it to the numerous resources that have joined it lately in the NLP resource box, as can be seen in Open Linked Data[1] – a large graph connecting information from various resources.

[*] This material is based in part upon work supported by the National Science Foundation CAREER award #0747340. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[1] linkeddata.org

The standard graphs – consisting of a set of nodes and edges that connect pairs of nodes – have quickly grown into more powerful representations such as heterogeneous graphs, hypergraphs, graphs with multi-layered edges, to fit more and more complex problems or data, and to support computational approaches.

With a proper choice of nodes and edge drawing criteria and weighing, graphs can be extremely useful for revealing regularities and patterns in the data, allowing us to bypass the bottleneck of data annotations. Graph formalisms have been adopted as an unsupervised learning approach to numerous problems – such as language identification, part-of-speech (POS) induction, or word sense induction – and also in semi-supervised settings, where a small set of annotated seed examples are used together with the graph structure to spread their annotations throughout the graph. Graphs' appeal is also enhanced by the fact that using them as a representation method can reveal characteristics and be useful for human inspection, and thus provide insights and ideas for automatic methods.

All is not perfect in the world of graphs, however. Many graph-based algorithms are NP-hard, and do not scale to current data sizes. As a well-studied field in mathematics, there are proofs that the graph problems encountered converge, or have a solution. Finding it computationally is another issue altogether, and scalability is an important attribute for algorithms, as they have to process larger and larger amounts of data. There are also problems that pertain specifically to computational approaches in NLP – for example, streaming graphs – graphs that change (some of them very fast) in time, like the graphs built from social media, where the networks representing the users, their tweets and the relations between them change rapidly.

This all shows that graph construction is a critical issue – its structure must correctly model the data such that it will allow not only to solve the target NLP problem, but to solve it in a computationally acceptable manner (finite, and as reduced as possible, use of computation time and memory).

In this paper, we aim to present a broad overview of the status of graphs in NLP. We will focus in particular on the graph representations adopted, and show how the NLP task was mapped onto a graph-based problem. To cover as many different approaches as possible, we will not go into details that are not strictly connected to graphs. The included references are all available for exploring in more detail the approaches that the readers find most interesting.

Note that we focus on core NLP tasks, and will not delve into research topics that do not have a major NLP component (for example link prediction in social networks). We do not include descriptions of resources represented as graphs (e.g., WordNet, conceptual graphs). We also do not include graph methods used in sequence analysis, such as HMMs and related frameworks.

2 Notations and definitions

A *graph* $G = (V, E)$ is a structure consisting of a set of vertices (or nodes) $V = \{v_i | i = 1, n\}$, some of which are connected through a set of edges $E = \{(v_i, v_j) | v_i, v_j \in V\}$. In a weighted graph $G_w = (V, E, W)$, edges have associated a weight or cost w_{ij}:

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

$W = \{w_{ij} | w_{ij}$ is the weight/cost associated with edge $(v_i, v_j), w_{i,j} \in \mathbb{R}\}$. Edges can be directed or undirected.

Depending on the NLP application, the nodes and edges may represent a variety of language-related units and links. Vertices can represent text units of various sizes and characteristics, e.g., words, collocations, word senses, sentences or even documents. Edges can encode relationships like co-occurrence (two words appearing together in a text unit), collocation (two words appearing next to each other or separated by a function word), syntactic structure (e.g., the parent and child in a syntactic dependency), lexical similarity (e.g., cosine between the vector representations of two sentences).

In a *heterogeneous graph* the vertices may correspond to different types of entities, and the edges to different types of links between vertices of the same or different type: $V = V_1 \cup V_2 \cup \cdots \cup V_t$, where each V_i is the set of nodes representing one type of entity.

An example of a heterogeneous graph is a graph consisting of articles, their authors and bibliographic references. Edges between authors could correspond to *co-authorship/collaboration, citation*, edges between authors and their papers represent *authorship*, and links between two papers could represent *citation/reference* relations.

A *hypergraph* expands the notion of graph by having edges – called *hyperedges* – that cover an arbitrary number of vertices: $E = \{E_1, \ldots, E_m\}$ with $E_k \subseteq V, \forall k = 1, m$. When $|E_k| = 2, \forall k = 1, m$ the hypergraph is a standard graph (Gallo *et al.* 1993). The incidence matrix $A(n \times m) = [a_{ik}]$ of a hypergraph associates each row i with vertex v_i and each column k with hyperedge E_k. $a_{ik} = 1$ if $v_i \in E_k$.

A directed hypergraph has directed hyperedges, which are represented as ordered pairs $E_k = (X_k, Y_k)$, where X_k, Y_k are disjoint subsets of vertices, possibly empty. X_k is the *head* of E_k ($H(E_k)$), and Y_k is the *tail* ($T(E_k)$). The incidence matrix of the hypergraph can encode directionality:

$$a_{ik} = \begin{cases} -1 & \text{if } v_i \in H(E_k) \\ 1 & \text{if } v_i \in T(E_k) \\ 0 & \text{otherwise} \end{cases}$$

An example of a hypergraph in language is the grammar, where the nodes are nonterminals and words, and each hyperedge corresponds to a grammatical rule, with the left-hand side of the rule forming the head of the hyperedge, and the body of the rule forming the tail.

3 Books and surveys

The most comprehensive book on the topic is Mihalcea and Radev (2011), which gives an introduction to graph theory, and presents in detail algorithms particularly relevant to various aspects of language processing, texts and linguistic knowledge as networks, and the combination of the two leading to elegant solutions for information retrieval, various problems related to lexical semantics (synonym detection, word sense disambiguation, semantic class detection, semantic distance), syntax (POS tagging, dependency parsing, prepositional phrase attachment), discourse

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

(co-reference resolution), as well as high-end applications like summarization, segmentation, machine translation.

Graphs and graph-based algorithms are particularly relevant for unsupervised approaches to language tasks. Choosing what the vertices represent, what their features are, and how edges between them should be drawn and weighted, leads to uncovering salient regularities and structure in the language or corpora data represented. Such formalisms are detailed in Biemann (2012), with emphasis on the usefulness of the graph framework to tasks superficially very different: language separation, POS tagging, word sense induction and word sense disambiguation. At the bottom of all these varied tasks is the phenomenon of clustering, for which the graph representation and algorithms are particularly appropriate. Chen and Ji (2010) present a survey of clustering approaches useful for tasks in computational linguistics.

Transforming a graph representation allows different characteristics of the data to come into focus – for example imposing a certain threshold on the weights of edges in a graph will change the topology of the structure, leading to different results in clustering. Rossi *et al.* (2012) examine and categorize techniques for transforming graph-based relational data – transformation of nodes/edges/features – to improve statistical relational learning. Rossi *et al.* present a taxonomy for data representation transformation in relational domains that incorporates link transformation and node transformation as symmetric representation tasks. Relational representation transformation is defined as any change to the space of links, nodes and/or features used to represent the data. The particular transformation applied depends on the application, and may lead to improving the accuracy, speed or complexity of the final application – e.g., adding links between similar nodes may increase performance in classification/clustering. Transformation tasks for both nodes and links include (i) predicting their existence, (ii) predicting their label or type, (iii) estimating their weight or importance, (iv) constructing their relevant features.

Some of the most used techniques in graph-based learning approaches include min-cut (Blum and Chawla 2001), spectral graph transducer (Joachims 2003), random walk-based approaches (Szummer and Jaakkola 2001), and label propagation (Zhu and Ghahramani 2002). Label propagation in particular is frequently used: it is a method of self-supervision, by allowing the labels of a small set of annotated data to spread in consistent fashion (according to the underlying similarity method) to unlabeled data.

4 Text structure, discourse, and generation

While traditionally we work with clean, edited text, the increasing amounts and the appeal of data produced through social media (like Tweeter and Facebook) raises the need for text normalization and typo correction to provide clean data to NLP tools further down the processing chain. This section reviews a few approaches that address this issue with graph-based methods.

Once a clean text is obtained, a potential next step is inducing its structure, to detect semantically coherent segments. This structuring can further aid tasks such

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

as summarization and alignment. The idea that a summary should consist of the most semantically rich and representative sentences of a document has led to the development of approaches that aim to detect simultaneously the keyphrases and the most important sentences of a document/set of documents. Graph representations can capture this duality, and bipartite or heterogeneous graphs have been used to model both keyphrase and sentence nodes, and the relations between them. Keyphrases are themselves a desired result, as they can contribute to document classification or clustering.

Texts also have a discourse structure, whether they are a simple text or a multi-party dialog. The set of entity mentions in the text and the coreference relations between them can themselves be modeled through different graph representations, either to make local decisions about a pair of entity mentions, or to induce clusters representing coreference chains that group all mentions of an entity together.

4.1 Text normalization

The language of social media is very dynamic, and alternative spellings (and errors) for words based on ad-hoc or customized abbreviations, phonetic substitutions or slang language are continuously created. Text normalization could be used to increase the performance of subsequent processing such as Machine Translation, Text-to-Speech, Information Extraction. Hassan *et al.* (2013) proposed a method that relies on a method similar to label propagation – from correct word forms found in dictionaries – to alternative spellings. This approach relies on a bipartite graph $G = (W, C, E)$ which represents words $W = \{w_i | i = 1, N\}$ and contexts $C = \{C_j | j = 1, M\}$ which are n-gram patterns. Frequency-based weighted edges connect words with the contexts in which they appear. The graph is built based on social media noisy text, and a large clean corpus. Correct words are marked based on frequency information from the clean corpus. These are the 'correctly labeled' words. Unlabeled nodes will adopt the 'label' (i.e., spelling) of the their closest (highest ranking) labeled node based on a random walk in graph G.

Interaction with social media from portable devices like smartphones brings up particular problems for languages with logographic scripts, like Chinese, as the small screen cannot display the hundreds of available characters. The solution is the usage of input method engines (IME) of which pinyin-to-Chinese conversion is a core part. This manages the conversion from (Roman alphabet) letter sequences to logograms (Chinese characters), but is prone to errors on two levels: (i) the sequence of letters input has a typo – caused by limited familiarity with the language or dialect, or mistake – and the system cannot produce the correct character, (ii) the wrong Chinese character was selected for the correct input letters. Jia and Zhao (2014) address these problems through a combination of two graph-based methods. The first method is applied to a graph consisting of the linear sequence of letters input by the user, and aims to produce legal syllables (as sequences of nodes) that have a corresponding Chinese character using a dictionary. Each detected syllable will form a new node, and it will be connected to other adjacent candidate syllables. A new graph will be built based on the detected syllables candidates, plus syllables

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

that are similar to these candidates based on a Levenshtein distance. The shortest path that covers the string is taken as the best typo correction result. To determine the correct mapping onto Chinese characters, an HMM is applied to the sequence of typo-corrected syllables, as each syllable can be mapped onto different characters.

4.2 Text structure and summarization

Despite the fact that often when reading a text we intuitively detect functional structures – an introduction, the elaboration/main content, a conclusion – texts often have at most a structuring in terms of paragraphs that may or may not reflect a shared topic of the sentences included.

Among the first to explore the structure of a text computationally through graph-based methods, Salton *et al.* (1997) apply techniques previously used to determining inter-document link to determine links between sentences or paragraphs within a document. The weights of the edges are essentially similarity scores between the nodes, filtered using a threshold value. The first aim of the work is to determine the structure of a text as a sequence of coherent units. This emerges when edges are further limited to connect nodes corresponding to sentences or paragraphs no more than five positions away. Summarization is an extension of the analysis of the text structure in terms of segments. They propose that this structure of segments can be used to produce the generic summary of a text by selecting a subset of the sentences/paragraphs that cover all the topics of the document. Three methods are explored, based on the 'bushiness' of nodes – what current graph formalisms call the degree of nodes. The best performing was the 'bushy path' method, that selected the top k bushy nodes, where k is the targeted number of paragraphs in the summary.

Zha (2002) proposes a new graph representation of a document based on the intuition that important terms and sentences should reinforce each other. Instead of linking together sentences through an edge representing the similarity of the two, Zha differentiates between sentences and keyphrases, and build an undirected bipartite graph that captures the occurrence of keyphrases in sentences. The aim is to score each node in this graph based on its links and the weights of these links, and this score will be the 'salience' of the node. The scores of the nodes are computed in a manner very similar to the HITS algorithm (Kleinberg 1999), where the keyphrases and sentences are scored iteratively depending on each others' scores until convergence. This approach determines a ranking of keyphrases (and sentences) that can be used to describe the document. The next step is to leverage this information to build a summary. The first operation is to cluster sentences. The weight of an edge between two sentences depends on the number and weight of the keyphrases they share. Recognizing that the order in which sentences appear is important, the weight of the edge has an additional (fix) factor (α) which is added when two sentences 'are near-by' or not. To cluster the sentences, spectral clustering is applied to the incidence matrix of the sentence graph. This is used to produce a hierarchical clustering of sentences. Depending on the level of summarization (more detailed or more general), clusters at different levels can be used, and then representative sentences selected from each cluster.

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

SNo	ID	Text
1	d1s1	Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.
2	d2s1	Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq rejects cooperating with the United Nations except on the issue of lifting the blockade imposed upon it since the year 1990.
3	d2s2	Ramadan told reporters in Baghdad that "Iraq cannot deal positively with whoever represents the Security Council unless there was a clear stance on the issue of lifting the blockade off of it.
4	d2s3	Baghdad had decided late last October to completely cease cooperating with the inspectors of the United Nations Special Commission (UNSCOM), in charge of disarming Iraq's weapons, and whose work became very limited since the fifth of August, and announced it will not resume its cooperation with the Commission even if it were subjected to a military operation.
5	d3s1	The Russian Foreign Minister, Igor Ivanov, warned today, Wednesday against using force against Iraq, which will destroy, according to him, seven years of difficult diplomatic work and will complicate the regional situation in the area.
6	d3s2	Ivanov contended that carrying out air strikes against Iraq, who refuses to cooperate with the United Nations inspectors, "will end the tremendous work achieved by the international group during the past seven years and will complicate the situation in the region."
7	d3s3	Nevertheless, Ivanov stressed that Baghdad must resume working with the Special Commission in charge of disarming the Iraqi weapons of mass destruction (UNSCOM).
8	d4s1	The Special Representative of the United Nations Secretary-General in Baghdad, Prakash Shah, announced today, Wednesday, after meeting with the Iraqi Deputy Prime Minister Tariq Aziz, that Iraq refuses to back down from its decision to cut off cooperation with the disarmament inspectors.
9	d5s1	British Prime Minister Tony Blair said today, Sunday, that the crisis between the international community and Iraq "did not end" and that Britain is still "ready, prepared, and able to strike Iraq."
10	d5s2	In a gathering with the press held at the Prime Minister's office, Blair contended that the crisis with Iraq "will not end until Iraq has absolutely and unconditionally respected its commitments" towards the United Nations.
11	d5s3	A spokesman for Tony Blair had indicated that the British Prime Minister gave permission to British Air Force Tornado planes stationed in Kuwait to join the aerial bombardment against Iraq.

Fig. 1. A cluster of 11 related sentences.

In Erkan and Radev (2004) and Mihalcea and Tarau (2004), they take the idea of graph-based summarization further by introducing the concept of *lexical centrality*. Lexical centrality is a measure of importance of nodes in a graph formed by linking semantically or lexically related sentences or documents. A random walk is then executed on the graph and the nodes that are visited the most frequently are selected as the summary of the input graph (which, in some cases, consists of information from multiple documents). One should note however, that in order to avoid nodes with duplicate or near duplicate content, the final decision about including a node in the summary also depends on its maximal marginal relevance as defined in Carbonell and Goldstein (1998). An example from Erkan and Radev (2004) is shown in Figure 1. The input consists of eleven sentences from several news stories on related topics. Figure 2 shows the resulting weighted graph.

To boost scores for the most relevant or important sentences, the sentence-based graph representations for documents can be enhanced with additional information such as relative position of sentences within a document (Wan 2008). Word-based graph representations could include POS information, sentences in which they occurred and position in these sentences. Ganesan, Zhai and Han (2010) use such a representation, in which words are linked based on their sequence in the sentence (adjacent words are connected with directed edges). Three properties of this graph – redundancy; gapped subsequence; collapsible structure – are used to explore and

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

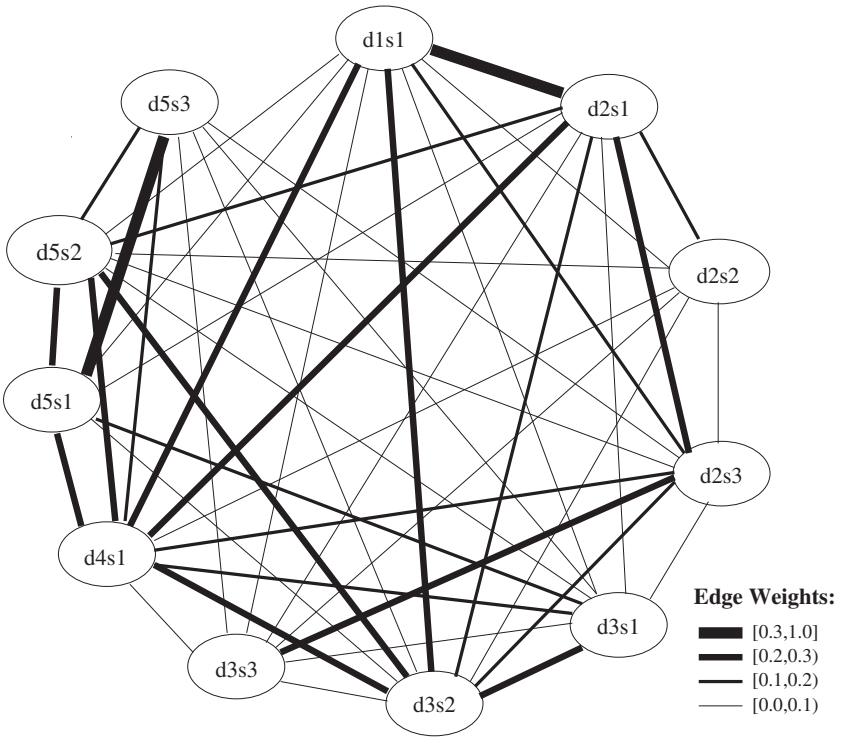

Fig. 2. Weighted cosine similarity graph for the cluster in Figure 1.

score subpaths that help generating abstractive summaries (as they have elements of sentence fusion and compression based on the selected paths).

Zhu *et al.* (2013) formulate the informative-sentence problem in opinion summarization as a community-leader detection problem, where a community consists of a cluster of sentences towards the same aspect of an entity. The graph consists of sentences linked by edges whose weight combines term similarity and adjective orientation similarity. In this graph, an interactive process builds communities of sentences and determines their leaders: a set of leaders is selected initially (from the top nodes based on their degree, select a set of k nodes such that no two are connected), then iteratively communities and leaders are updated in a manner similar to link propagation: starting with the current set of leaders the communities are determined (one per leader), and after generating the community, leaders are reassigned based on ranking their in-community degree. After the process converges, a set of informative sentences are selected from each community to generate the summary.

A different approach to summarization is presented by Mani and Bloedorn (1997), who start with the goal of building summaries from several documents. They build a graph for each document, whose nodes are word instances at specific positions (such that names and phrases spanning more than one word are formed by the respective word instances). Words are weighted and filtered using tf-idf, and are connected through several types of relations, presented in order of weight: *same* (linking

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

different instances of the same word); *coreference* (link names or phrases that are the same – names and phrases span more than one word/node); *name* (link nodes that together form a name); *phrase* (link nodes that are part of the same phrase); *alpha* (various lexical relations such as synonymy, hypernymy/hyponymy obtained from an early version of WordNet); *adj* (adjacency – words that are adjacent in the text, but filtering out intervening words). On this graph with weighted nodes and edges, is applied a step of spreading activation. First, a set of words expressing a topic of interest is selected. All nodes in the graph except those matching the topic words receive a weight of 0. Starting from the selected topic words the spreading activation will reassign weights to the nodes, based on the signal coming from connected nodes, the weight of the edges, and a dampening effect caused by distance from a starting node. After activation, segments are selected from the reweighted graph. A segment can either consist of the set of nodes (and underlying text) within a weight within a given delta from the peak values, or all nodes within a user-defined distance in the text from a peak value.

Spreading activation for topic-driven summarization was also used by Nastase (2008). The set of documents is used to build a graph in which open-class words are nodes connected through dependency relations. In this graph, open words from the topic and their related terms obtained from Wikipedia and WordNet are given a starting weight which is then propagated using spreading activation to enhance the weight of other related terms and the edges that connect them. The weighted nodes and relations are used to score the sentences in which they appear, and the highest scoring ones will form the summary.

Related to the problem of summarization is the issue of passage retrieval: given a query in the form of a natural language question, return a set of passages from a set of input documents that contain the answer. Otterbacher, Erkan and Radev (2005) propose a solution that combines the sentence-based graph representation of Erkan and Radev (2004) with biased random walk and implement a label propagation method: the graph is seeded with known positive and negative examples and then each node is labeled in proportion to the percentage of times a random walk on the graph ends at that node. Given the presence of the initially labeled nodes, the nodes with the highest score eventually are the ones that are both similar to the seed nodes and are central to the document set. In other words, they are chosen as the answer set by a mixture model that takes into account the known seeds (positive or negative) and the lexical centrality score as in the previous section. The graph consists of both sentences (paragraphs) and features (content words that appear in these sentences). The graph is bipartite as a sentence can only link to a feature and vice versa.

4.3 Discourse

Coreference resolution aims to group mentions of entities such that all mentions in a group refer to the same entity. This problem can be cast into a graph-based framework in various ways. For instance, Ng (2009) uses a graph formalism to filter non-anaphoric mentions, as previous work has shown that eliminating isolated

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

mentions (singletons) leads to better coreference resolution results. The solution proposed partitions a graph in two parts corresponding to anaphoric and non-anaphoric mentions. The graph's nodes are the mentions discovered in the text, plus two special nodes – s (source) and t (sink) – representing the two classes (anaphoric/non-anaphoric). The graph is built in two steps. First, each mention node n is connected to the s and t nodes through edges whose weights are a function of the probability that n is anaphoric or not. In the next step, mention nodes n_i and n_j are connected through an edge weighted by a similarity measure between n_i and n_j, reflecting the probability that the two are coreferent. Partitioning this graph in two is a minimum cut problem, which seeks to minimize the partition cost, i.e., the cost of 'cut' edges, where the nodes they link belong to the different subsets. Training data is used to estimate probabilities and thresholds on these probabilities for weighing/drawing the graph.

Other approaches aim to cluster the mentions based on connections between them. (Nicolae and Nicolae 2006) build a graph whose vertices are mentions, connected with edges whose weights are confidence values obtained from a coreference classification model. This graph is then partitioned into clusters using a variation of the min-cut algorithm that iteratively removes edges between subgraphs that have low weights, and are thus interpreted as representing different entities.

Cai and Strube (2010) present a one-step coreference method that builds coreference chains directly by clustering nodes in a hypergraph. Hypergraph nodes are mentions detected in the text, and the edges group nodes that can be connected through relational features (e.g., *alias* – the mentions are aliases of each other: proper names with partial match, full names and acronyms or organizations, etc.; *synonyms*; etc.) The edges of the hypergraph correspond roughly to features used in other coreference work. This hypergraph covering the mentions in the entire document is split into sub-hypergraphs (i.e., clusters) by partitioning using two-way recursive spectral clustering.

Local text coherence can also be cast into a graph framework. Occurrence of entities in sentences can be viewed as a bipartite graph, and used to model local coherence (Guinaudeau and Strube 2013). Links between entities and sentences can encode grammatical information (e.g., entity is subject/object in the sentence), and be weighted accordingly. This bipartite graph is used to generate sentence graphs, where two sentences are connected if they have at least one entity in common. Depending on how the weights of the graph are computed, several variants of the sentence graphs are obtained. Compared to alternative approaches for sentence ordering, summary coherence rating and readability assessment, the graph-based approach is computationally lighter at state-of-the-art performance levels.

Another discourse problem is dialog analysis, of which disentanglement – determining to which conversation thread each utterance belongs to – is an essential step. Elsner and Charniak (2010) approach this as a clustering problem on a graph. A machine learning step is first used to predict probabilities for pairs of utterances as belonging to the same conversation thread or not based on lexical, timing and discourse-based features. The graph covering the conversation is then built, with a node for each utterance, and edges between utterances having as weight a function

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

of the probability score assigned by the classifier (the log odds). On this graph is applied a greedy voting algorithm, adding an utterance j to an existing cluster based on the weight of the edge between j and nodes in the existing cluster, or put it into a new cluster if no weights greater than 0 exist.

4.4 Language generation

From the point of view of graphs, paraphrases can be seen as matching graphs – there is a mapping between the graphs (as dependency graphs or syntactic trees) corresponding to the paraphrases. Barzilay and Lee (2003) build word latices to find commonalities within automatically derived groups of structurally similar sentences. They then identify pairs of lattices from different corpora that are paraphrases of each other – the identification process checks whether the lattices take similar arguments; given an input sentence to be paraphrased, they match it to a lattice and use a paraphrase from the matched lattice's mate to generate an output sentence.

Konstas and Lapata (2012) generate descriptions of database records in natural language. Given a corpus of database records and textual descriptions (for some of them), they define a PCFG that captures the structure of the database and how it can be rendered into natural language. This grammar, representing a set of trees, is encoded as a weighted hypergraph. Generation is equivalent to finding the best derivation tree in the hypergraph using Viterbi.

5 Syntax and tagging

Regarding syntax, we have identified two main directions – graphs used to represent the dependency relations between words, and graphs for representing the grammar, used ultimately in generative contexts (in machine translation or language generation).

Tagging involves assigning (one of the given) tags to words or expressions in a collection. Approaches using graphs rely on the fact that they are useful for providing a global view on the data and enforce coherence at the level of the entire dataset. This characteristic is exploited to induce consistent labeling over a set of nodes, either by clustering, propagating the tags starting from a small set of seeds, or by obtaining features that capture a larger context of the targeted entity for supervised learning.

5.1 Syntactic parsing

Dependency relations linking words in a sentence form a directed acyclic graph. This view of the result of syntactic parsing can be used to cast the problem of dependency parsing into searching for a maximum spanning tree (MST) in a directed graph that covers the given sentence/text (Hirakawa 2001; McDonald *et al.* 2005): given a directed graph $G = (V, E)$, the MST problem is to find the highest scoring subgraph of G that satisfies the tree constraint over the set of vertices V.

Graph literature provides various algorithms for determining the MST of a directed graph. Choosing an algorithm depends on characteristics of the dependency

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

graph: for projective dependencies[2] choose one based on the Eisner algorithm (Eisner 1996); for non-projective dependencies choose one based on Chi-Liu-Edmonds (Chu and Liu 1965; Edmonds 1967).

Another important aspect is scoring the MST candidates. There are several variations, based on the way the scoring of the tree is done: *first-order* – the score of the tree is based on the scores of single edges; *second-order* – the score of the tree is factored into the sum of adjacent edge-pair scores.

Graph-based models take into account the score for the entire structure, but this score is computed based on local features of each edge, to make the parsing tractable. Nivre and McDonald (2008), Zhang and Clark (2008) and Chen and Ji (2010) show methods to improve the graph-based parsing by including additional features, possibly produced by alternative parsing models. Nivre and McDonald (2008) and Zhang and Clark (2008) use features produced by transition models – learned by scoring transitions from one parser state to the next – which have a complementary approach to parsing compared to the graph-based models – they use local training, and greedy inference algorithms, while using richer features that capture the history of parsing decisions. It is interesting to note that the transition-based and the graph-based parsing have the same end states – the set of dependency relations graphs that cover the input sentence – which they reach through different search strategies. Combining features that guide the search strategies for the two methods leads to improved results.

The definition of directed hyperarcs in terms of heads and tails matches the view of grammatical rules – which have a head and a body, and therefore can be used to encode (probabilistic) grammars (Klein and Manning 2001). Building a hypergraph that encodes a grammar and an input, the paths in the hypergraph correspond to parses of the given input. The shortest path will correspond to the best parse. Klein and Manning (2001) present PCFG-specific solutions in the hypergraph framework, including an approach that constructs the grammar hypergraph dynamically as needed, and a Dijkstra's algorithm style shortest path computation. Other solutions were proposed by Huang and Chiang (2005) and Huang (2008), which can also be integrated in the decoding step of phrase-based or syntax-based machine translation (Huang and Chiang 2007), where grammar rules are combined with language models.

5.2 Tagging

Using graph methods for tagging relies on the intuition that similar entities should have the same tag. The nodes in these graph will represent words or phrases (depending on the type of targets and their tags), and the edges will be drawn and weighted based on a similarity metric between the nodes.

Watanabe, Asahara and Matsumoto (2007) aim to tag named entities in Wikipedia. A graph structure covers linked anchor texts of hyperlinks in structured portions in Wikipedia articles – in particular lists and tables. A CRF variation is used to categorize nodes in the graph as one of twelve Named Entity types. Three types of

[2] A word and its descendants form a contiguous substring of the sentence.

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

links are defined between anchor texts, based on their relationships in the structured portions of the text – siblings, cousins, relatives. These relations define three types of cliques. The potential function for cliques is introduced to define conditional probability distribution over CRFs (over label set y given observations x). These potential functions are expressed in terms of features that capture co-occurrences between labels. Experiments show that a configuration using cousin and relative relations leads to the best results (also compared to a non-graph method – i.e., unconnected nodes).

Subramanya, Petrov and Pereira (2010) tag words with POS information through a label-propagation algorithm that builds upon a word similarity graph and the assumption that words that are similar have the same POS. The similarity graph is used during the training of a CRF to smooth the state posteriors on the target domain. Local sequence contexts (n-grams) are graph vertices, exploiting the empirical observation that the POS of a word occurrence is mostly determined by its local context. For each n-gram they extract a set of context features, whose values are the pointwise mutual information between the n-gram and its features. The similarity function between graph nodes is the cosine distance between the pointwise mutual information vectors representing each node. The neighbors of a node are used as features for the CRF, thus embedding larger contextual information in the model. CRFs cannot enforce directly constraints that similar n-grams appearing in different contexts should have similar POS tags. The graphs are used to discover new features, to propagate adjustments to the weights of known features, and to train the CRF in a semi-supervised manner.

Bollegala, Matsuo and Ishizuka (2008) detect aliases based on a word (anchor text) co-occurrence graph in which they compute node rankings, combined using SVMs. The nodes consist of words that appear in anchor texts, which are linked through an edge if the anchor texts in which they appear point to the same URL. The association strength between a name and a candidate alias is computed using several measures (link frequency – the number of different URLs in which the name and candidate co-occur), tf-idf (to downrank high frequency words), log-likelihood ratio, chi-squared measure, pointwise mutual information and hypergeometric distribution), also considering the importance of each URL target.

6 Semantics

Within the area of lexical and text semantics, the most common representation is a graph having words as nodes. The way edges are drawn and weighted varies much, depending on the task. They may represented directed/undirected relations, and may be derived from other networks (e.g., as similarity/distance from WordNet), from distributional representations of words, or directly from evidence found in corpora (e.g., corresponding to conjunctions of the form *X (and—or—but) Y*).

The purpose of the tasks also varies. The focus may be to build a lexical network, to transfer annotations from one lexical network to another, or to induce higher level information, such as semantic classes or even ontologies.

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

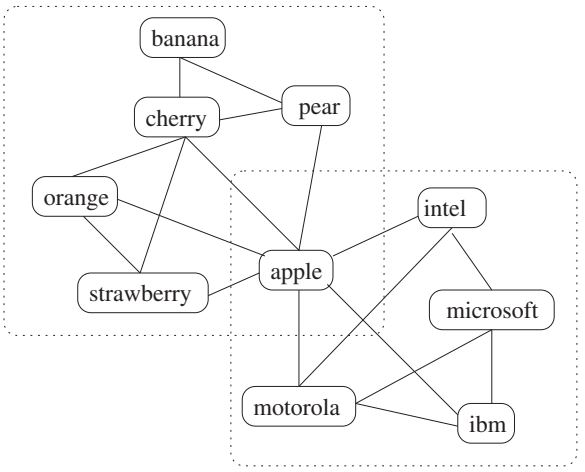

Fig. 3. Lexical network constructed for the extraction of semantic classes.

6.1 Lexicon and language models

One of the largest graph representations constructed to support an NLP task is perhaps the graph model proposed by Widdows and Dorow for unsupervised lexical acquisition (Widdows and Dorow 2002). The goal of their work is to build semantic classes, by automatically extracting from raw corpora all the elements belonging to a certain semantic category such as *fruits* or *musical instruments*. The method first constructs a large graph consisting of all the nouns in a large corpus (British National Corpus, in their case), linked by the conjunction *and* or *or*. A cutoff value is used to filter out rare words, resulting in a graph of almost 1,00,000 nouns, linked by more than half-million edges. To identify the elements of a semantic class, first a few representative nouns are manually selected and used to form a seed set. In an iterative process, the node found to have the largest number of links with the seed set in the co-occurrence graph is selected as potentially correct, and thus added to the seed set. The process is repeated until no new elements can be reliably added to the seed set. Figure 3 shows a sample of a graph built to extract semantic classes. An evaluation against ten semantic classes from WordNet indicated an accuracy of 82 per cent which, according to the authors, was an order of magnitude better than previous work in semantic class extraction. The drawback of their method is the low coverage which is limited to those words found in a conjunction relation. However, whenever applicable, the graph representation has the ability to precisely identify the words belonging to a semantic class.

Another research area is the study of lexical network properties carried out by Ferrer-i-Cancho and Sole (2001). By building very large lexical networks of nearly half-million nodes, with more than ten million edges, constructed by linking words appearing in English sentences within a distance of at most two words, they proved that complex system properties hold on such co-occurrence networks. Specifically, they observed a small-world effect, with a relatively small number of 2–3 jumps required to connect any two words in the lexical network. Additionally, it has also been observed that the distribution of node degrees inside the network is scale-free,

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

which reflects the tendency of a link to be formed with an already highly connected word. Perhaps not surprisingly, the small-world and scale-free properties observed over lexical networks automatically acquired from corpora are were also observed on manually-constructed semantic networks such as WordNet (Sigman and Cecchi 2002; Steyvers and Tenenbaum 2005).

In a more recent work on acquiring semantic classes and their instances, Talukdar *et al.* (2008) use a graph formalism to encode information from unstructured and structured texts and then induce and propagate labels. Nodes representing instances or classes are extracted from free text using clustering techniques and structured sources (like HTML tables). A small set of nodes is annotated with class labels (which also appear as class nodes in the graph), and these labels are propagated in the graph using Adsorption label propagation, which computes for each node a probability distribution over the set of labels. Talukdar and Pereira (2010) continues this work by comparing several label propagation algorithms for this problem, determining that Modified Adsorption gives the best results. Modified Adsorption is a variation of the Adsorption algorithm, formalized like an optimization problem.

Velardi, Faralli and Navigli (2013) learn concepts and relations via automated extraction of terms, definitions and hypernyms to obtain a dense hypernym graph. A taxonomy is induced from this (potentially disconnected and cyclic) graph via optimal branching and weighting.

As seen above, corpus information can be exploited to obtain structured information. One downside of information derived from corpora is the fact that it captures information at the word level and connecting to other linguistic resources such as WordNet or FrameNet requires word-sense distinctions. Johansson and Nieto Piña (2015) present a framework for deriving vector representations for word senses from continuous vector-space representations of the words and word sense information (and their connections) from a semantic network. The work is based on word-sense constraints in the semantic network – neighbors in the semantic network should have similar vector representations – and the fact that the vector for a polysemous word is a combination of the vectors of its senses.

Numerous lexical resources, including those automatically derived, have a graph structure. To combine such resources Matuschek and Gurevych (2013) iteratively determine an alignment using the graphs representing these resources and an initial set of trivial alignments consisting of monosemous nodes in both resources. Further alignments are based on the shortest path in the connected graph that links a pair of candidate nodes, one from each of the initial resources.

From monolingual lexical networks we can transition to multi-lingual networks by linking together monolingual networks. Issues like inducing new connections starting from a seed of relations that link the networks, and disambiguating ambiguous entries are seamlessly tackled in the graph-based framework. Laws *et al.* (2010) build separate graphs for two languages, representing words and their lexical relations (e.g., adjectival modification). The two monolingual graphs are linked starting with a set of seeds. Nodes from the two graphs are compared and linked using a similarity measure to determine translations. Flati and Navigli (2012) disambiguate ambiguous translations in the lexical entries of a bilingual machine-readable dictionary using

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

cycles and quasi-cycles. The dictionary is represented as a graph and cyclic patterns are sought in this graph to assign an appropriate sense tag to each translation in a lexical entry. The output is also used to correct the dictionary by improving alignments and missing entries.

6.2 Similarity and relatedness measures

A large class of methods for semantic similarity consists of metrics calculated on existing semantic networks such as WordNet and Roget, by applying, for instance, shortest path algorithms that identify the closest semantic relation between two input concepts (Leacock, Miller and Chodorow 1998). Tsatsaronis, Varlamis and Nørvåg (2010) present a method for computing word relatedness based on WordNet that exploits several types of information in the network: depth of nodes, relations and relation weights, relations crossing POS boundaries. The computation is extended from word-to-word to relatedness between texts.

Hughes and Ramage (2007) propose an algorithm based on random walks. Briefly, in their method, the PageRank algorithm is used to calculate the stationary distribution of the nodes in the WordNet graph, biased on each of the input words in a given word pair. Next, the divergence between these distributions is calculated, which reflects the relatedness of the two words. When evaluated on standard word relatedness data sets, the method was found to improve significantly over previously proposed algorithms for semantic relatedness. In fact, their best performing measure came close to the upper bound represented by the inter-annotator agreement on these data sets.

Tsang and Stevenson (2010) introduce a measure of the semantic distance between texts that integrates distributional information with a network flow formalism. Texts are represented as a collection of frequency weighted concepts within an ontology. The network flow method provides an efficient way of explicitly measuring the frequency-weighted ontological distance between concepts across two texts.

A different approach to similarity computation that combines co-occurrence information from a parsed corpus is presented by Minkov and Cohen (2008). The starting point is a graph with two types of vertices and two types of edges that covers a dependency parsed corpus: nodes are work tokens and word types (terms), edges representing grammatical dependencies connect word token vertices, the inverse relation is then added, and there are also edges linking word tokens with the corresponding word type (term). The working assumption is that terms that are more semantically related will be linked by a larger number of paths in this corpus graph, and shorter paths are more meaningful. The similarity between two nodes in this graph is derived through a weighted random walk. The edges may have uniform weights, or they can be tuned in a learning step. For specific tasks, additional information from the graph can be used to rerank the terms with the highest similarity to terms in the given query (for example) – the sequence of edges on the connecting path, unigrams that appear on the path, and the number of words in the query that are connected to the term that is being ranked. Minkov and Cohen also propose a dynamic version of graph-walk, which is constrained at each

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

new step by previous path information. This is achieved by reevaluating the weights of the outgoing edges from the current edge based on the history of the walk up to this node.

6.3 Word sense induction and word sense disambiguation

The surface level of a text consists of words, but what a reader perceives, and what we'd ideally want a system to access, are the meanings of words, or word senses. It is commonly accepted that the context of a word – within a window of a given size/sentence/larger text fragment – influences its interpretation and thus determines its sense. Mapping words onto specific senses can be done relative to a given inventory of senses, or a system may determine itself the set of senses that occur in a given text collection, or something in between when a partial set of senses can be provided for a small set of seed words. Depending on the task and the availability of labeled data, various graph-based methods can be applied, including clustering on unlabeled data, label propagation starting from a small set of labeled data, ranking of given word senses to determine which applies to specific instances in the data.

Work related to word senses has been encouraged by recurring word sense induction and word sense disambiguation tasks within the SensEval/SemEval/*SEM semantic evaluation campaigns. The variety of approaches has been recorded in the events' proceedings. We will present an overview of graph-based methods successfully used to tackle these tasks by modeling the relations between words, their contexts and their senses, and using these models in different manners.

A graph-based method that has been successfully used for semi-supervised word sense disambiguation is the label propagation algorithm (Niu, Ji and Tan 2005). In their work, Niu and colleagues start by constructing a graph consisting of all the labeled and unlabeled examples provided for a given ambiguous word. The word sense examples are used as nodes in the graph, and weighted edges are drawn by using a pairwise metric of similarity. On this graph, all the known labeled examples (the seed set) are assigned with their correct labels, which are then propagated throughout the graph across the weighted links. In this way, all the nodes are assigned with a set of labels, each with a certain probability. The algorithm is repeated through convergence, with the known labeled examples being reassigned with their correct label at each iteration. In an evaluation carried out on a standard word sense disambiguation data set, the performance of the algorithm was found to exceed the one obtained with monolingual or bilingual bootstrapping. The algorithm was also found to perform better than SVM when only a few labeled examples were available.

Graph-based methods have also been used for knowledge-based word sense disambiguation. In Mihalcea, Tarau and Figa (2004), Mihalcea and colleagues proposed a method based on graphs constructed based on WordNet. Given an input text, a graph is built by adding all the possible senses for the words in the text, which are then connected on the basis of the semantic relations available in the WordNet lexicon. For instance, Figure 4 shows an example of a graph constructed over a short sentence of four words.

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

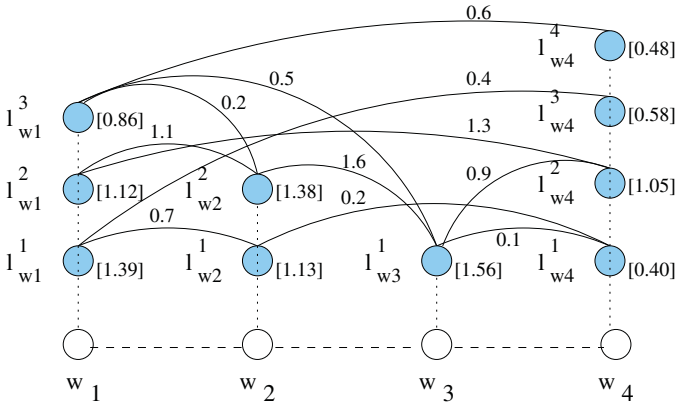

Fig. 4. (Colour online) Graph constructed over the word senses in a sentence, to support automatic word sense disambiguation.

A random-walk applied on this graph results in a set of scores that reflects the 'importance' of each word sense in the given text. The word senses with the highest score are consequently selected as potentially correct. An evaluation on sense-annotated data showed that this graph-based algorithm was superior to alternative knowledge-based methods that did not make use of such rich representations of word sense relationships.

In follow-up work, Mihalcea developed a more general graph-based method that did not require the availability of semantic relations such as those defined in WordNet. Instead, she used derived weighted edges determined by using a measure of similarity among word sense definitions (Mihalcea 2005), which brought generality, as the method is not restricted to semantic networks such as WordNet but it can be used on any electronic dictionaries, as well as improvements in disambiguation accuracy.

Along similar lines with (Mihalcea *et al.* 2004), Navigli and Lapata carried out a comparative evaluation of several graph connectivity algorithms applied on word sense graphs derived from WordNet (Navigli and Lapata 2007). They found that the best word sense disambiguation accuracy is achieved by using a closeness measure, which was found superior to other graph centrality algorithms such as in-degree, PageRank, and betweenness. Navigli and Lapata (2010) present an updated survey of graph-based methods for word sense disambiguation. Agirre, de Lacalle and Soroa (2014) present a random walk-based disambiguation method on a combination of WordNet and extended WordNet. Extended WordNet (Mihalcea and Moldovan 2001) brings in relations between synsets and disambiguated words in the synset glosses. This additional information makes the graph more dense, which leads to better results of the PageRank algorithm for word sense disambiguation than WordNet alone.

In the related task of entity linking – essentially disambiguating a named entity relative to an inventory of possible interpretations/concepts – Fahrni, Nastase and Strube (2011) starts from an n-partite graph similar to Mihalcea *et al.* (2004), where each part corresponds to the possible interpretations of the corresponding text mention. Edges between potential interpretations are weighted based on a

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

combination of relatedness measures that capture relatedness information between these interpretations from Wikipedia (if they can be mapped onto a Wikipedia article), as well as context selectional preference. Concepts are then chosen using a maximum edge weighted clique algorithm – choose the interpretations that have the highest scored subgraph. The method achieved highest scores in the NTCIR-9 entity linking task for several languages (Japanese, Korean, Chinese) and evaluation methods. For the same task, Moro, Raganato and Navigli (2014) use a densest subgraph heuristic together with entity candidate meanings to select high-coherence semantic interpretations. The graph consists of terms in the texts and their candidate meanings, whose edges are reweighted using random walks and triangles. The highest density subgraph heuristic provides the joint disambiguation solution.

Graph connectivity can also be used to tackle the complementary problem of word sense induction. Word sense induction is often modeled as a clustering problem, with word occurrences – represented through their contexts – that share the same word sense grouped together. Graph-based word sense induction rely usually on the co-occurrence graph, where (open-class, or just nouns) words are nodes. Nodes corresponding to words that occur together within a pre-specified span (e.g., document, sentence, or a specific window size) are connected with edges whose weights reflect co-occurrence frequency, pointwise mutual information between the two words, or other co-occurrence measures. The assumption is that clusters in this network will correspond to different word senses (Biemann 2012). Nodes could also represent word pairs (target word,collocate) to better separate subgraphs pertaining to different senses of the same target word. Nodes are weighted based on the frequency of the corresponding word pair, and nodes that come from the same context are connected (Klapaftis and Manandhar 2008). Clustering using the Chinese whispers algorithm proceeds iteratively, with vertices all assigned to different classes, and then reassigned at every step based on the strongest class in its local neighborhood (Biemann 2012). Building the graph relies on several parameters, that threshold and weight the nodes and edges. Korkontzelos, Klapaftis and Manandhar (2009) explore eight graph connectivity measures that evaluate the connectivity of clusters produced by a graph-based word sense induction method based on a set of parameters. The evaluation allows the system to estimate the sets of parameters that lead to high performance. Di Marco and Navigli (2013) investigate the effect of different similarity measures used to draw and weigh edges in a word-based co-occurrence graph.

The previously mentioned approaches to word sense disambiguation either pair a target word with its collocates within the same node, or connects two co-occurring words together. Different models of the problem are proposed in Klapaftis and Manandhar (2007) and Qian *et al.* (2014), who use hypergraphs – actually hyperedges – to capture shared semantic context. Klapaftis and Manandhar (2007) build a hypergraph where nodes are words, and hyperedges connect words within the same context. In Qian *et al.* (2014)'s hypergraph, the nodes represent instances of the context where a target word appears, hyperedges represent higher-order semantic relatedness among these instances – particularly lexical chains. This representation captures a more global perspective as different contexts can be connected through

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

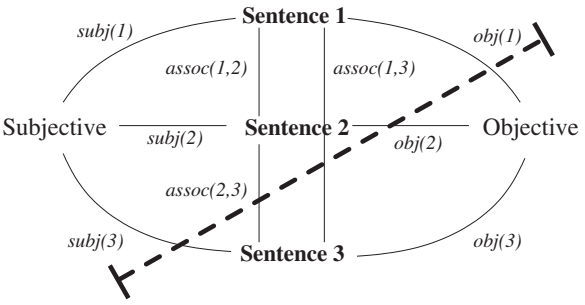

Fig. 5. A min-cut algorithm applied on a graph constructed over the sentences in a text, which is used to separate subjective from objective sentences.

a lexical chain. To induce word senses (Klapaftis and Manandhar 2007; Qian *et al.* 2014) use hypergraph clustering methods such as Normalized Hypergraph Cut (Zhou, Huang and Schölkopf 2006), Hyperedge Expansion Clustering (Shashua, Zass and Hazan 2006), or a maximal density clustering algorithm (Michoel and Nachtergaele 2012).

For processing semantic roles, Lang and Lapata (2014) represent argument instances of a verb as vertices in a graph whose edges express similarities between these instances. The graph consists of multiple edge layers, each capturing a different aspect of argument-instance similarity. This graph is partitioned based on extensions of standard clustering algorithms.

7 Sentiment analysis and social networks

Sentiment and subjectivity analysis is an area related to both semantics and pragmatics, which has received a lot of attention from the research community. An interesting approach based on graphs has been proposed by Pang and Lee (2004), where they show that a min-cut graph-based algorithm can be effectively applied to build subjective extracts of movie reviews.

First, they construct a graph by adding all the sentences in a review as nodes, and by drawing edges based on sentence proximity. Each node in the graph is initially assigned with a score indicating the probability of the corresponding sentence being subjective or objective, based on an estimate provided by a supervised subjectivity classifier. A min-cut algorithm is then applied on the graph and used to separate the subjective sentences from the objective ones. Figure 5 illustrates the graph constructed over the sentences in a text, on which the min-cut algorithm is applied to identify and extract the subjective sentences.

The precision of this graph-based subjectivity classifier was found to be better than the labeling obtained with the initial supervised classifier. Moreover, a polarity classifier relying on the min-cut subjective extracts was found to be more accurate than one applied on entire reviews.

Recent research on sentiment and subjectivity analysis has also considered the relation between word senses and subjectivity (Wiebe and Mihalcea 2006). In work targeting the assignment of subjectivity and polarity labels to WordNet senses, Esuli and

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

Sebastiani applied a biased PageRank algorithm on the entire WordNet graph (Esuli and Sebastiani 2007). Similar to some extent to the label propagation method, their random-walk algorithm was seeded with nodes labeled for subjectivity and polarity. When compared to a simpler classification method, their random-walk was found to result in more accurate annotations of subjectivity and polarity of word senses.

One of the first methods of inducing the semantic orientation of words is Hatzivassiloglou and McKeown (1997). They build a graph of adjectives, and draw edges based on conjunctions found in corpora, following the observation that if they appear in a conjunctions, the adjectives will have the same orientation (e.g., 'happy and healthy'). Adjectives are clustered based on the connectivity of the graph, and those in the same cluster will have the same label, thus expanding from an initial set of labeled seeds.

Graph methods for semantic orientation rely on a graph of words, seeded with semantic orientation information for a small subset of the nodes. The edges are drawn based on a variety of similarity metrics, relying on lexical resources (such as WordNet) or distributional representation from a corpus or the Web. Inducing the labels of unlabeled nodes is done in various manners such as label propagation (Blair-Goldensohn *et al.* 2008; Rao and Ravichandran 2009; Velikovich *et al.* 2010), or random walks (Xu, Meng and Wang 2010; Hassan *et al.* 2014). Blair-Goldensohn *et al.* (2008) and Rao and Ravichandran (2009) apply the label propagation algorithm on a graph built based on WordNet's synonymy and antonymy links. Velikovich *et al.* (2010) apply a variation on the label propagation algorithm (which considers only the highest scoring path from a labeled node to an unlabeled one) on a large graph of n-grams built based on the information in four billion pages. Context vectors and cosine similarity were used to draw edges. Hassan *et al.* (2014) apply random walks from unlabeled nodes to labeled ones, and estimate the orientation of the unlabeled nodes based on its relative proximity to positive/negative words. Xu *et al.* (2010) use random walks for ranking words based on the seed words. The method can be applied on a multilingual graph, to transfer sentiment information from one language to another through random walks (Hassan *et al.* 2014) or label propagation (Gao *et al.* 2015).

Semantic orientation can be transferred between languages using graph alignments. Scheible *et al.* (2010) build monolingual sentiment graphs for the source and target language respectively, and then align nodes in the two graphs based on a similarity measure that relies on the topology of each graph and a set of seed links between them, as in the SimRank algorithm (Jeh and Widom 2002; Dorow *et al.* 2009). The influence of different phenomena (coordinations through 'and' and 'but', adjective-noun modification) can be computed separately and then averaged to obtain the final similarity score for two compared nodes. Similar nodes will have similar orientation. Gao *et al.* (2015) present a similar approach, building a graph consisting of two monolingual subgraphs for the source and target languages respectively. The link between the two graphs consists of an inter-language subgraph that connects the two based on word alignment information in a parallel corpus. The edges in the monolingual subgraphs can have positive or negative weights, corresponding to synonymy/antonymy relations between the words. Label

propagation is used to propagate sentiment polarity labels from the source language (English) to the target language (Chinese).

To build a tweet recommendation system that presents users with items they may have an interest in, Yan, Lapata and Li (2012) build a heterogeneous graph, which is used to rank both tweeters and tweets simultaneously. This graph covers the network of tweeters, the network of tweets linked based on content similarity, and includes additional edges that link these two based on posting and retweeting information. Nodes are ranked based on coupling two random walks, one on the graph representing the tweeters, the other the tweets. The framework was also extended to allow for personalized recommendations, by ranking tweets relative to individual users.

Another interesting task related to social media is determining the polarity of the users and the content they produce. Zhu *et al.* (2014) build a tripartite graph with the purpose of determining the polarity of tweets and tweeters. The graph nodes represent users, their tweets, and features of the users and the tweets (as words in the user profile and in the tweets). Edges between user nodes and tweet nodes represent posting or retweeting, and feature nodes are linked to the user and tweet nodes with which they appear. Co-clustering in this graph will produce simultaneously sentiment clusters of users, tweets and features. Recognizing that such graphs change fast over time, leads to an online setting where an initial graph is updated with new network information (new users, tweets and features), which allows them to study the dynamic factor of user-level sentiments and the evolution of latent feature factors.

8 Machine translation

Label propagation approaches are based on the smoothness assumption (Chapelle, Schölkopf and Zien 2006) which states that if two nodes are similar according to the graph, their output labels should also be similar. We have seen in previous sections the label propagation algorithm – which usually relies on a small set of labels (e.g., binary) that will be propagated – applied to text normalization, passage retrieval, semantic class acquisition, word sense induction and disambiguation, semantic orientation. The goal of the label propagation algorithm is to compute soft labels for unlabeled vertices from the labeled vertices. The edge weight encodes (intuitively) the degree of belief about the similarity of the soft labeling for the connected vertices.

Labels to be propagated need not be atomic, but can also be 'structured' – e.g., the label is a translation of the node's string. In this format, the technique can be applied to machine translation, particularly to encourage smooth translation probabilities for similar inputs.

The first machine translation approach using graph-based learning is presented by Alexandrescu and Kirchhoff (2009). They build a graph consisting of train and test data (word strings) connected through edges that encode pairwise similarities between samples. The training data will have labels – i.e., translations – that will be propagated to the unlabeled data based on the similarity relations between nodes.

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

Label options for unlabeled nodes (i.e., candidate translations) are first produced using an SMT system, and the label propagation algorithm is used to rerank the candidates, ensuring that similar nodes (i.e., input strings) will have similar labels. The similarity measure used to compute edge weights is crucial to the success of the method, and can be used to incorporate domain knowledge. Alexandrescu and Kirchhoff compare two measures – the BLEU score (Papineni *et al.* 2002) and a score based on string kernels. On different datasets, different similarity measures perform better. An important issue facing graph-based learning is scalability, because the working graph combines training and test data. To address this issue, a separate graph is built for each test sentence, as a transitive closure of the edge set over the nodes containing all hypotheses for that test sentence. A similar approach is presented in Liu *et al.* (2012).

One of the causes of errors in machine translation are out-of-vocabulary words. Razmara *et al.* (2013) use label propagation to find translations (as labels) for out-of-vocabulary words. A graph is constructed from source language monolingual texts, and the source side of the available parallel data. Each phrase type represents a vertex in the graph, and is connected to other vertices with a weight defined by a similarity measure between the two profiles (and filtered based on a threshold value). There are three types of vertices: labeled, unlabeled, and out-of-vocabulary. Nodes for which translations are available (from the parallel data/phrase tables) are annotated with target-side translations and their feature values. A label propagation algorithm is used to propagate translations form labeled nodes to unlabeled nodes. This handles several types of out-of-vocabulary words, including morphological variants, spelling variants and synonyms. The graph constructed is very large, the experiments show that the methods proposed are scalable.

Graphs can be used to combine different translation models in one structure, where the models can complement or strengthen each other's choices. Cmejrek, Mi and Zhou (2013) introduce the 'flexible interaction of hypergraphs' where translation rules from a tree-to-string and hierarchical phrase-based model are combined in a hypergraph, which is then used for decoding. Tree-to-string translation rules – consisting of a tree fragment on the left-hand side, and a string on the right-hand side in the target language – are considered to be good at non-local reorderings, while hierarchical phrase-based rules – consisting of a source-language string on the left-hand side and a target-language string on the right – are good at providing reliable lexical coverage. The hypergraph is built from these rules: left and right sides of these rules will become nodes with an associated span (start and end point in the source or target language string). Nodes from different rules that cover the same span are merged – forming *interaction supernodes*. Nodes within an interaction supernode are connected through *interaction edges*. Interaction hyperedges within each supernode allow the decoder to switch between models.

9 Information extraction/Knowledge extraction and representation/Events

Information extraction and representation is a multi-faceted problem, and this is reflected in the variety of graph-based approaches proposed. One characteristic of

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

the problem which makes it particularly appropriate for a graph approach is the redundancy in the data – the same type of information can appear in numerous contexts or forms. Redundancy can be explored to boost particular patterns – as vertices or edges or paths within the representation graph.

For identifying topics of a given document, Coursey and Mihalcea (2009) use high ranking nodes in a very large graph built based on Wikipedia articles and categories, scored through a biased graph centrality algorithm started from Wikipedia concepts identified in the input document. Several variations regarding the structure of the graph are tested, with the best performance obtained from a graph that has as nodes both Wikipedia articles and categories. Within the ranking process, the best performing bias takes into account the nodes in the graph that have been identified in the input document (through a wikification process).

A popular approach to information extraction is bootstrapping – start with a few seed relation examples or patterns, and iteratively grow the set of relations and patterns based on occurrence in a large corpus (Hearst 1992). This view of bootstrapping as a mutual dependency between patterns and relation instances can be modeled through a bipartite graph. Hassan, Hassan and Emam (2006) cast the relation pattern detection as a hubs (instances) and authorities (patterns) problem, solved using the HITS algorithm (Kleinberg 1999). The method relies on redundancy in large datasets and graph-based mutual reinforcement to induce generalized extraction patterns. The mutual reinforcement between patterns and instances will lead to increased weight for authoritative patterns, which will then be used for information extraction. To reduce the space of instances and induce better patterns, instances are clustered based on a similarity/relatedness measure based on WordNet between the entities in the same position in a pair of instances.

Bootstrapping algorithms are prone to semantic drift – where patterns that encode relations different that the target one are started to be extracted, which leads to the extraction of noisy instances, which in turn lead to more noisy patterns, and so on. Komachi *et al.* (2008) show that semantic drift observed in bootstrapping algorithms is essentially the same phenomenon as topic drift in the HITS algorithm through an analysis of HITS-based algorithm performance in word sense disambiguation. Comparison of the ranking of instances (text fragments containing a target word) obtained through bootstrapping and the HITS algorithm show that the two methods arrive at the same results. To address the issue of semantic drift, they propose two graph-based algorithms (von-Neumann kernels and regularized Laplacian), for scoring the instances relative to the patterns, which will keep the extraction algorithm more semantically focused.

While semantic networks and ontologies that include knowledge about words/word senses/concept as a hierarchy are quite common, similar knowledge structures that encode relations between larger text units are just starting to appear. One such knowledge structure is an *entailment graph* (Berant, Dagan and Goldberger 2010). The entailment graph is a graph structure over propositional templates, which are propositions comprising a predicate and arguments, possibly replaced by variables – e.g., *alcohol reduces blood pressure, X reduces Y*. Berant *et al.* (2010) present a global algorithm for learning entailment relations between propositional templates.

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

The optimal set of edges is learned using Integer Linear Programming – they define a global function and aim to find the graph that maximizes the function under a transitivity constraint.

Representing temporal interactions between events in a text is another problem where graphs are a good fit. The problem is how to build them. Bramsen *et al.* (2006) compare three different approaches to building directed acyclic that encode temporal relations found in texts. They are all based on predictions for pairs of events (edges) – forward, backward, null – learned from manually annotated data. These local decisions though can be combined in different ways to arrive at the big picture. From the three methods investigated – (i) Natural Reading Order – start with an empty graph and add the highest scoring edge for a new node (event) that appears in text without violating the consistency of the direct acyclic graph; (ii) Best-First – add edges to obtain the highest scoring graph, by always adding the highest scoring edge that doesn't violate the direct acyclic graph condition; (iii) exact inference with Integer Linear Programming – build a globally optimal temporal direct acyclic graph as an optimization problem, subject to the following constraints: there is exactly one relation (edge) between two events (nodes), the transitivity constraint is respected, and the direct acyclic graph is connected. The graph construction method using Integer Linear Programming provides the best results.

Events have multiple facets, e.g., the outcome, its causes, aftermath. To detect the facets of an event and group together blog posts about a facet of the same event, Muthukrishnan, Gerrish and Radev (2008) use KL divergence and the Jaccard coefficient to generate topic labels (as keyphrases) and then build a topic graph which represents the community structure of different facets. The graph built has keyphrases as nodes, linked with edges weighted with an overlap measure (Jaccard similarity coefficient, defined as a ratio of the documents covered by both keyphrases and the total number of documents covered by the two keyphrases). A greedy algorithm is used to iteratively extract a Weighted Set Cover using a cost function for each node (i.e., keyphrase) that combines coverage information and coverage overlap with other keyphrases.

Popular application areas for event extraction are the medical and biological domains, to help find and aggregate data from an ever increasing number of studies. To find events and their arguments in biological texts, Björne *et al.* (2009) represent texts as semantic graphs – entities and events connected by edges corresponding to event arguments.

Notions like minimal graphs of a graph are useful for casting a difficult evaluation problem into a manageable formalism. Evaluation of NLP problems can be difficult – e.g., the evaluation of temporal graphs that capture temporal relations between events in text. Allen's relations (seven direct + six inverse) have been adopted for annotation of events' temporal relations. Evaluating the annotations against a gold standard is difficult because the level of the relations may vary: the same ordering of events may be expressed in different ways, or they may include relation closures that may artificially increase a score. Tannier and Muller (2011) propose a method to address these issues and provide an objective evaluation metric. First,

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

based on fact that the Allen relations are defined in terms of the ends of the time interval corresponding to an event, they transform the graph where events are nodes connected by the Allen relations into a graph where the nodes are the start and end points of events, and the relations between them can be equality, before or after. From this graph, the 'gold standard' reference graph is extracted as the *minimal graph* of constraints. A minimal graph has the following two properties: (1) its (relation) closure leads to the full graph; (2) removing any relation leads to breaking the first property. Minimal graphs of a candidate temporal events annotation can be compared to the reference minimal graph objectively.

10 Further reading

General graph and network analysis papers. The following papers describe the relevant graph theory: (Doyle and Snell 1984; Bollobás 1985; Bollobás 1998; Brin and Page 1998; Grimmett 1999; Langville and Meyer 2003). **Lexical networks**. The following readings are essential: (Dorogovtsev and Mendes 2001; Motter *et al.* 2002; de Moura, Lai and Motter 2003; Ferrer i Cancho 2005; Caldeira *et al.* 2006; Masucci and Rodgers 2006; Pardo *et al.* 2006; Ferrer i Cancho *et al.* 2007), and (Mehler 2007). **Language processing applications**. A list includes (Haghighi, Ng and Manning 2005; Wolf and Gibson 2005; Zens and Ney 2005; Erkan 2006; Malioutov and Barzilay 2006), and (Biemann 2006). **Random walks and learning on graphs**. Some readings include (Zhu and Ghahramani 2002; Radev 2004; Zhu and Lafferty 2005; Goldberg and Zhu 2006), and (Zhu 2007).

The lists above are by far not exhaustive. A large bibliography appears on Dragomir Radev's web site `http://clair.si.umich.edu/~radev/webgraph/webgraph-bib.html`.

11 Conclusions

In this paper, we presented an overview of the current state-of-the-art in research work on graphs in NLP. We addressed the relevant work in the main areas of NLP, including text structure and discourse, semantics and syntax, summarization and generation, machine translation, and information and knowledge extraction. We covered both the graph representations used to model the problems, as well as the graph algorithms applied on these representations. We believe the intersection of the fields of natural language processing and graph theory has proven to be a rich source of interesting solutions that has just been untapped. We expect that future work in this space will bring many more exciting findings.

References

Agirre, E., de Lacalle, O. L., and Soroa, A. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* **40**(1): 57–84.

Alexandrescu, A., and Kirchhoff, K. 2009. Graph-based learning for statistical machine translation. In *Proceedings of Human Language Technologies 2009: The Conference of the North American Chapter of the Association for Computational Linguistics,* Boulder, Col., 31 May–5 June 2009, pp. 119–127.

https://doi.org/10.1017/S1351324915000340 Published online by Cambridge University Press

Barzilay, R., and Lee, L. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Edmonton, Alberta, Canada, 27 May –1 June 2003.

Berant, J., Dagan, I., and Goldberger, J. 2010. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pp. 1220–1229.

Biemann, C. 2006. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the Second Workshop on Graph Based Methods for Natural Language Processing,* New York City, NY, USA, June 2006, pp. 73–80.

Biemann, C. 2012. *Structure Discovery in Natural Language*. Berlin/Heidelberg: Springer.

Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., and Salakoski, T. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the BioNLP 2009 Workshop, Companion Volume for Shared Task*, pp. 10–18.

Blair-Goldensohn, S., Neylon, T., Hannan, K., Reis, G. A., McDonald, R., and Reynar, J. 2008. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era (NLPIX)*.

Blum, A., and Chawla, S. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning,* Williams College, Williamstown, MA, USA, June 28–July 1 2001, pp. 19–26.

Bollegala, D., Matsuo, Y., and Ishizuka, M. 2008. A co-occurrence graph-based approach for personal name alias extraction from anchor texts. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing,* Hyderabad, India, 7–12 January 2008, pp. 865–870.

Bollobás, B. 1985. *Random Graphs*. London, UK: Academic Press.

Bollobás, B. 1998. *Modern Graph Theory*. New York: Springer.

Bramsen, P., Deshpande, P., Lee, Y. K., and Barzilay, R. 2006. Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,* Sydney, Australia, 22–23 July 2006, pp. 189–198.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* **30**(1–7): 107–117.

Cai, J., and Strube, M. 2010. End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pp. 143–151.

Caldeira, S. M. G., Petit Lobão, T. C., Andrade, R. F. S., Neme, A., and Miranda, J. G. V. 2006. The network of concepts in written texts. *European Physical Journal B* **49**(4): 523–529, February.

Carbonell, J. G., and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336.

Chapelle, O., Schölkopf, B., and Zien, A. (eds.) 2006. *Semi-Supervised Learning*. Adaptive computation and machine learning. Cambridge, MA, USA: MIT Press.

Chen, C., and Ji, H. 2010. Graph-based clustering for computational linguistics: a survey. In *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-5, pp. 1–9.

Chu, Y. J., and Liu, T. H. 1965. On the shortest arborescence of a directed graph. *Science Sinica* **14**: 1396–1400.

Cmejrek, M., Mi, H., and Zhou, B. 2013. Flexible and efficient hypergraph interactions for joint hierarchical and forest-to-string decoding. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing,* Seattle, USA, 18–21 October 2013, pp. 545–555.

Coursey, K., and Mihalcea, R. 2009. Topic identification using wikipedia graph centrality. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing,* Singapore, 2–7 August 2009, pp. 117–120.

de Moura, A. P. S., Lai, Y.-C., and Motter, A. E. 2003. Signatures of small-world and scale-free properties in large computer programs. *American Physical Society* **68**(1): 017102–1–017102–4, July.

Di Marco, A., and Navigli, R. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics* **39**(3): 710–754.

Dorogovtsev, S. N., and Mendes, J. F. F. 2001. Language as an evolving word Web. *Proceedings of the Royal Society of London B* **268**(1485): 2603–2606, December 22.

Dorow, B., Laws, F., Michelbacher, L., Scheible, C., and Utt, J. 2009. A graph-theoretic algorithm for automatic extension of translation lexicons. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pp. 91–95.

Doyle, P. G., and Snell, J. L. 1984. Random walks and electric networks. Technical Report math.PR/0001057, Arxiv.org.

Edmonds, J. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, **71B**: 233–240.

Eisner, J. M. 1996. Three new probabilistic models for dependency parsing: an exploration. In *Proceedings of the 16th International Conference on Computational Linguistics,* Copenhagen, Denmark, 5–9 August 1996, pp. 340–345.

Elsner, M., and Charniak, E. 2010. Disentangling chat. *Computational Linguistics* **36**(1): 389–409.

Erkan, G. 2006. Language model-based document clustering using random walks. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* New York, N.Y., 4–9 June 2006, pp. 479–486.

Erkan, G., and Radev, D. 2004. The university of Michigan at DUC 2004. In *Document Understanding Conference (DUC)*, Boston, Massachusetts, May.

Esuli, A., and Sebastiani, F. 2007. PageRanking WordNet synsets: an application to opinion mining. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* Prague, Czech Republic, 23–30 June 2007, pp. 424–431.

Fahrni, A., Nastase, V., and Strube, M. 2011. Hits graph-based system at the ntcir-9 cross-lingual link discovery task. In *Proceedings of the 9th NTCIR (NII Test Collection for IR Systems) Workshop Meeting,* Tokyo, Japan, 6–9 December 2011, pp. 473–480.

Fellbaum, C. (eds.) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Ferrer i Cancho, R. 2005. The structure of syntactic dependency networks: insights from recent advances in network theory. In V. Levickij, and G. Altmann (eds.), *Problems of Quantitative Linguistics*, pp. 60–75. Ruta.

Ferrer i Cancho, R., Mehler, A., Pustylnikov, O., and Díaz-Guilera, A. 2007. Correlations in the organization of large-scale syntactic dependency networks. In *TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*, pp. 65–72. Rochester, New York, USA: Association for Computational Linguistics.

Ferrer i Cancho, R., and Sole, R. V. 2001. The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences* **268**(1482): 2261–2265, November.

Flati, T., and Navigli, R. 2012. The cqc algorithm: cycling in graphs to semantically enrich and enhance a bilingual dictionary. *Journal of Artificial Intelligence Research* **43**: 135–171.

Gallo, G., Longo, G., Nguyen, S., and Pallotino, S. 1993. Directed hypergraphs and applications. *Discrete Applied Mathematics* **42**: 177–201.

Ganesan, K., Zhai, C. X., and Han, J. 2010. Opinosis: a graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pp. 340–348.

Gao, D., Wei, F., Li, W., Liu, X., and Zhou, M. 2015. Cross-lingual sentiment lexicon learning with bilingual word graph label propagation. *Computational Linguistics* **41**(1): 21–40.

Goldberg, A. B., and Zhu, J. 2006. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*, pp. 45–52.

Grimmett, G. 1999. *Percolation*, vol. 321, 2nd ed. Grundlehren der mathematischen Wissenschaften. Springer.

Guinaudeau, C., and Strube, M. 2013. Graph-based local coherence modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics,* Sofia, Bulgaria, 4–9 August 2013, pp. 93–103.

Haghighi, A., Ng, A., and Manning, C. 2005. Robust textual inference via graph matching. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pp. 387–394.

Hassan, A., Abu-Jbara, A., Lu, W., and Radev, D. R. 2014. A random walk-based model for identifying semantic orientation. *Computational Linguistics* **40**(3): 539–562.

Hassan, H., Hassan, A., and Emam, O. 2006. Unsupervised information extraction approach using graph mutual reinforcement. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,* Sydney, Australia, 22–23 July 2006, pp. 501–508.

Hassan, H., and Menezes, A. 2013. Social text normalization using contextual graph random walks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics,* Sofia, Bulgaria, 4–9 August 2013, pp. 1577–1586.

Hatzivassiloglou, V., and McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and of the 8th Conference of the European Chapter of the Association for Computational Linguistics,* Madrid, Spain, 7–12 July 1997, pp. 174–181.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics,* Nantes, France, 23–28 August 1992, pp. 539–545.

Hirakawa, H. 2001. Semantic dependency analysis method for Japanese based on optimum tree search algorithm. In *Proceedings of the 5th Meeting of the Pacific Association for Computational Linguistics,* Kitakyushu, Japan, 11–14 September 2001.

Huang, L. 2008. Forest reranking: discriminative parsing with non-local features. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* Columbus, Ohio, 15–20 June 2008, pp. 586–594.

Huang, L., and Chiang, D. 2005. Better k-best parsing. In *Proceedings of IWPT 2005*.

Huang, L., and Chiang, D. 2007. Forest rescoring: faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* Prague, Czech Republic, 23–30 June 2007, pp. 144–151.

Hughes, T., and Ramage, D. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning,* Prague, Czech Republic, 28–30 June 2007, pp. 581–589.

Jeh, G., and Widom, J. 2002. SimRank: a measure of structural-context similarity. In *Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* Edmonton, Alberta, Canada, 23–26 July 2002, pp. 538–543.

Jia, Z., and Zhao, H. 2014. A joint graph model for pinyin-to-chinese conversion with typo correction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* Baltimore, Maryland, USA, 22–27 June, 2014, pp. 1512–1523.

Joachims, T. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of the 18th International Conference on Machine Learning,* Williams College, Williamstown, MAWashington, D.C., USA, August 21–24 2003, pp. 290–297.

Johansson, R., and Nieto Piña, L. 2015. Embedding a semantic network in a word space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Denver, Colorado, USA, May 31–June 5 2015, pp. 1428–1433.

Klapaftis, I., and Manandhar, S. 2007. UOY: a hypergraph model for word sense induction and disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-1),* Prague, Czech Republic, 23–24 June 2007, pp. 414–417.

Klapaftis, I., and Manandhar, S. 2008. Word sense induction using graphs of collocations. In *Proceedings of the 18th European Conference on Artificial Intelligence,* Patras, Greece, 21–25 July 2008, pp. 298–302.

Klein, D., and Manning, C. D. 2001. Parsing and hypergraphs. In *Proceedings of the 7th International Workshop on Parsing Technologies*, Beijing, China, 17–19 October, pp. 123–134.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *JACM* **46**(5): 604–632, September.

Komachi, M., Kudo, T., Shimbo, M., and Matsumoto, Y. 2008. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 1011–1020.

Konstas, I. and Lapata, M. 2012. Unsupervised concept-to-text generation with hypergraphs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Montréal, Quebec, Canada, 3–8 June 2012, pp. 752–761.

Korkontzelos, I., Klapaftis, I., and Manandhar, S. 2009. Graph connectivity measures for unsupervised parameter tuning of graph-based sense induction systems. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, pp. 36–44, Boulder, Colorado, USA, June. Association for Computational Linguistics.

Lang, J., and Lapata, M. 2014. Similarity-driven semantic role induction via graph partitioning. *Computational Linguistics* **40**(3): 633–669.

Langville, A. N., and Meyer, C. D. 2003. Deeper inside PageRank. *Internet Mathematics* **1**(3): 335–380.

Laws, F., Michelbacher, L., Dorow, B., Scheible, C., Heid, U., and Schütze, H. 2010. A linguistically grounded graph model for bilingual lexicon extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pp. 614–622.

Leacock, C., Miller, G. A., and Chodorow, M. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics* **24**(1): 147–165.

Liu, S., Li, C.-H., Li, M., and Zhou, M. 2012. Learning translation consensus with structured label propagation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* Jeju Island, Korea, 8–14 July 2012, pp. 302–310.

Malioutov, I., and Barzilay, R. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics,* Sydney, Australia, 17–21 July 2006, pp. 25–32.

Mani, I., and Bloedorn, E. 1997. Multi-document summarization by graph searching and matching. In *Proceedings of the 14th National Conference on Artificial Intelligence,* Providence, R.I., 27–31 July 1997, pp. 622–628.

Masucci, A. P., and Rodgers, G. J. 2006. Network properties of written human language. *Physical Review E* **74**, August 2.

Matuschek, M., and Gurevych, I. 2013. Dijkstra-wsa: a graph-based approach to word sense alignment. *Transactions of the Association for Computational Linguistics* **1**(2013): 151–164.

McDonald, R., Pereira, F., Ribarov, K., and Hajic, J. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Human Language Technology*

Conference and the 2005 Conference on Empirical Methods in Natural Language Processing, Vancouver, B.C., Canada, 6–8 October 2005, pp. 523–530.

Mehler, A. 2007. Large text networks as an object of corpus linguistic studies. In A. Lüdeling and M. Kytö (eds.), *Corpus Linguistics. An International Handbook of the Science of Language and Society*, pp. 328–382. Berlin/New York: de Gruyter.

Michoel, T., and Nachtergaele, B. 2012. Alignment and integration of complex networks by hypergraph-based spectral clustering. *Physical Review E* **86**: 056111.

Mihalcea, R. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pp. 411–418.

Mihalcea, R., and Moldovan, D. 2001. eXtended WordNet: progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, pp. 95–100.

Mihalcea, R., and Radev, D. R. 2011. *Graph-Based Natural Language Processing and Information Retrieval.* Cambridge University Press.

Mihalcea, A., and Tarau, P. 2004. Textrank: bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing,* Barcelona, Spain, 25–26 July 2004, pp. 404–411.

Mihalcea, R., Tarau, R., and Figa, E. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics,* Geneva, Switzerland, 23–27 August 2004, pp. 1126–1132.

Minkov, E., and Cohen, W. W. 2008. Learning graph walk based similarity measures for parsed text. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 907–916.

Moro, A., Raganato, A., and Navigli, R. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics* **2**(2014): 231–244.

Motter, A. E., de Moura, A. P. S., Lai, Y.-C., and Dasgupta, P. 2002. Topology of the conceptual network of language. *Physical Review E* **65**(065102): 065102–1–065102–4, June 25.

Muthukrishnan, P., Gerrish, J., and Radev, D. R. 2008. Detecting multiple facets of an event using graph-based unsupervised methods. In *Proceedings of the 22nd International Conference on Computational Linguistics,* Manchester, UK, 18–22 August 2008, pp. 609–616.

Nastase, V. 2008. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 763–772.

Navigli, R., and Lapata, M. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence,* Hyderabad, India, 6–12 January 2007, pp. 1683–1688, Hyderabad, India.

Navigli, R., and Lapata, M. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(4): 678–692.

Ng, V. 2009. Graph-cut-based anaphoricity determination for coreference resolution. In *Proceedings of Human Language Technologies 2009: The Conference of the North American Chapter of the Association for Computational Linguistics,* Boulder, Col., 31 May–5 June 2009, pp. 575–583.

Nicolae, C., and Nicolae, G. 2006. Bestcut: a graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,* Sydney, Australia, 22–23 July 2006, pp. 275–283.

Niu, Z.-Y., Ji, D.-H., and Tan, C. L. 2005. Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics,* Ann Arbor, Mich., 25–30 June 2005, pp. 395–402.

Nivre, J., and McDonald, R. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* Columbus, Ohio, 15–20 June 2008, pp. 950–958.

Otterbacher, J., Erkan, G., and Radev, D. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pp. 915–922.

Pang, B. and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics,* Barcelona, Spain, 21–26 July 2004, pp. 271–278.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, Penn., 7–12 July 2002, pp. 311–318.

Alexandre, T., Pardo, S., Antiqueira, L., das Graças Volpe Nunes, M., Oliveira Jr., O. N., and da Fontoura Costa, L. 2006. Modeling and evaluating summaries using complex networks. In *Proceedings of Computational Processing of the Portuguese Language, the Seventh International Workshop (PROPOR-2006)*, Springer, pp. 1–10.

Qian, T., Ji, D., Zhang, M., Teng, C., and Xia, C. 2014. Word sense induction using lexical chain based hypergraph model. In *Proceedings of the 25th International Conference on Computational Linguistics,* Dublin, Ireland, 23–29 August 2014, pp. 1601–1611.

Quillian, M. R. 1968. Semantic memory. In M. Minsky (ed.), *Semantic Information Processing*. Cambridge, MA: MIT Press.

Radev, D. R. 2004. Weakly supervised graph-based methods for classification. Technical Report CSE-TR-500-04, University of Michigan. Department of Electrical Engineering and Computer Science.

Radev, D. R., and Mihalcea, R. 2008. Networks and natural language processing. *AI Magazine* **29**(3): 16–28.

Rao, D., and Ravichandran, D. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics,* Athens, Greece, 30 March–3 April 2009, pp. 675–682.

Razmara, M., Siahbani, M., Haffari, R., and Sarkar, A. 2013. Graph propagation for paraphrasing out-of-vocabulary words in statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics,* Sofia, Bulgaria, 4–9 August 2013, pp. 1105–1115.

Rossi, R. A., McDowell, L. K., Aha, D. W., and Neville, J. 2012. Transforming graph data for statistical relational learning. *Journal of Machine Learning Research* **45**(1): 363–441.

Salton, G., Singhal, A., Mitra, M., and Buckley, C. 1997. Automatic text structuring and summarization. *Journal of Information Processing and Management: an International Journal – Special issue: methods and tools for the automatic construction of hypertext* **33**(2): 193–207, March.

Scheible, C., Laws, F., Michelbacher, L., and Schütze, H. 2010. Sentiment translation through multi-edge graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pp. 1104–1112.

Shashua, A., Zass, R., and Hazan, T. 2006. Multi-way clustering using super-symmetric non-negative tensor factorization. In A. Leonardis, H. Bischof, and A. Pinz (eds.), *Computer Vision – ECCV 2006*, pp. 595–608. Lecture Notes in Computer Science. Berlin Heidelberg: Springer.

Sigman, M., and Cecchi, G. A. 2002. Global organization of the Wordnet lexicon. *Proceedings of the National Academy of Sciences of the United States of America* **99**(3): 1742–1747, February 5.

Steyvers, M., and Tenenbaum, J. B. 2005. Graph theoretic analyses of semantic networks: small worlds in semantic networks. *Cognitive Science* **29**(1): 41–78.

Subramanya, A., Petrov, S., and Pereira, F. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing,* Cambridge, MA, 9–11 Ocotber 2010, pp. 167–176.

Szummer, M., and Jaakkola, T. 2001. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, pp. 945–952.

Talukdar, P. P., and Pereira, F. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pp. 1473–1481.

Talukdar, P. P., Reisinger, J., Pasca, M., Ravichandran, D., Bhagat, R., and Pereira, F. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 582–590.

Tannier, X., and Muller, P. 2011. Evaluating temporal graphs built from texts via transitive reduction. *Journal of Artificial Intelligence Research* **40**(2011): 375–413.

Tsang, V., and Stevenson, S. 2010. A graph-theoretic framework for semantic distance. *Computational Linguistics* **36**(1): 32–69.

Tsatsaronis, G., Varlamis, I., and Nørvåg, K. 2010. Semanticrank: ranking keywords and sentences using semantic graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pp. 1074–1082.

Velardi, P., Faralli, S., and Navigli, R. 2013. Ontolearn reloaded: a graph-based algorithm for taxonomy induction. *Computational Linguistics* **39**(3): 665–707.

Velikovich, L., Blair-Goldensohn, S., Hannan, K., and McDonald, R. 2010. The viability of web-derived polarity lexicons. In *Proceedings of Human Language Technologies 2010: The Conference of the North American Chapter of the Association for Computational Linguistics,* Los Angeles, Cal., 2–4 June 2010, pp. 777–785.

Wan, X. 2008. An exploration of document impact on graph-based multi-document summarization. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 755–762.

Watanabe, Y., Asahara, M., and Matsumoto, Y. 2007. A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning,* Prague, Czech Republic, 28–30 June 2007, pp. 649–657.

Widdows, D., and Dorow, B. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics,* Taipei, Taiwan, 24 August–1 September 2002.

Wiebe, J., and Mihalcea, R. 2006. Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics,* Sydney, Australia, 17–21 July 2006, pp. 1065–1072.

Wolf, F., and Gibson, E. 2005. Representing discourse coherence: a corpus-based study. *Computational Linguistics* **31**(2): 249–288, June.

Xu, G., Meng, X., and Wang, H. 2010. Build chinese emotion lexicons using a graph-based algorithm and multiple resources. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pp. 1209–1217.

Yan, R., Lapata, M., and Li, X. 2012. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* Jeju Island, Korea, 8–14 July 2012, pp. 516–525.

Zens, R., and Ney, H. 2005. Word graphs for statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, Michigan, June. Association for Computational Linguistics, pp. 191–198.

Zha, H. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *SIGIR2002*, pp. 113–120.

Zhang, Y., and Clark, S. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 562–571.

Zhou, D., Huang, J., and Schölkopf, B. 2006. Learning with hypergraphs: clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, pp. 1601–1608.

Zhu, L., Galstyan, A., Cheng, J., and Lerman, K. 2014. Tripartite graph clustering for dynamic sentiment analysis on social media. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD-2014),* Snowbird, UT, USA, June 22–27, 2014, pp. 1531–1542.

Zhu, L., Gao, S., Pan, J., Li, H., Deng, D., and Shahabi, C. 2013. Graph-based informative-sentence selection for opinion summarization. In *Proceedings of the International Conference on Advances in Social Network Analysis and Mining (ASONAM-2013),* Niagara, Ontario, Canada, August 25–29, 2013, pp. 408–412.

Zhu, X. 2007. Semi-supervised learning literature survey. Technical Report TR 1530, Computer Sciences, University of Wisconsin, Madison.

Zhu, X., and Ghahramani, Z. 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.

Zhu, X., and Lafferty, J. 2005. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning,* Bonn, Germany, 7–11 August 2005, pp. 1052–1059.