Graph-Based Methods for Natural Language Processing and Understanding—A Survey and Analysis

Michael T. Mills and Nikolaos G. Bourbakis, *Fellow, IEEE*

***Abstract*—This survey and analysis presents the functional components, performance, and maturity of graph-based methods for natural language processing and natural language understanding and their potential for mature products. Resulting capabilities from the methods surveyed include summarization, text entailment, redundancy reduction, similarity measure, word sense induction and disambiguation, semantic relatedness, labeling (e.g., word sense), and novelty detection. Estimated scores for accuracy, coverage, scalability, and performance are derived from each method. This survey and analysis, with tables and bar graphs, offers a unique abstraction of functional components and levels of maturity from this collection of graph-based methodologies.**

***Index Terms*—Graph methods, natural language processing (NLP), natural language understanding (NLU).**

I. Introduction

THE vastness of information combined with the need for quick access to specific but comprehensive information has driven natural language processing (NLP) and natural language understanding (NLU) research to provide the following capabilities: event resolution (ER), grammar annotation (GrA), information mining (IM), knowledgebase (K), labeling (Lab), novelty detection (ND), question/answer (QA), redundancy reduction (Red), semantic relatedness (SR), similarity measure (SM), summarization (Sum), textual entailment (TE), word sense disambiguation (WSD), and word sense induction (WSI). Over the past ten years, research in these areas has moved toward graph-based methods. The reduced complexity of graph methods over vector methods offers a more compressed and efficient concept representation of text. This paper presents a summary of such graph-based methods that are found in recent technical publications plus an analysis of their component functions and their maturity calculated from information found in the referenced papers. The goal of this survey–analysis is to provide the reader with enough detailed information along with tables and charts to capture the current state of the art in graph-based methods for NLP and NLU, including their component functions, performance, and maturity. We conclude with an estimate of their near term potential to transform the results of this research into products.

The following describes each of these research areas as NLP and NLU capabilities. 1) Sum captures the main meanings of one or more documents down to a certain level of detail, or threshold. 2) TE, at the syntactic level, replaces all subsets of a text, with the encompassing text. At a semantic level, the concepts that represent the text subsets are replaced with one larger, encompassing concept without losing any meaning of the original text subsets. The desired result is a shorter summary with no redundancy and no loss of meaning from the original texts. 3) By measuring the similarity of text segments or their corresponding concepts, the resulting SM can be used to merge clusters of similar concepts into a single concept. This compresses the resulting, summarized text without losing any of the representing concepts of the original text segments. 4) WSI identifies words with the same meaning. This meaning identification can increase the accuracy of SM and yield a smaller text for summaries with less redundancy. 5) WSD uses context around sentences to reduce or eliminate ambiguous sentences that are caused by words having multiple meanings. Once words with the same or similar meaning are identified using WSI, WSD can use all the words with the same meaning within a context of other words to reduce ambiguous text even further. The relation (or association) of two or more words within a context is often referred to as coreference, i.e., the making of multiple instances of the same entity in a text. 6) SR is the relation between concepts. Concepts, relations, and attributes are used to represent text at a semantic level and higher level abstractions of the meanings represented by a body of text. A potential goal for such higher level (or semantic) representation is natural language understanding (NLU). 7) Lab is normally used to annotate parts of speech and senses (representing meanings) of words within a text. Lab can be generated by manual, supervised, unsupervised, and semisupervised methods. Supervised methods are trained using a manually labeled corpus (collection of text). Unsupervised methods are trained with a much larger collection of text. Semisupervised methods are trained using a relatively small labeled corpus to derive labels for a much larger unlabeled corpus. These learning methods are used, within this context, to determine senses of words in an unlabeled text. 8) ND uses some of the aforementioned capabilities to detect events of interest within a text.

The tables in this paper include the following information: 1) functional components from each methodology and resulting capability; 2) definitions of items that are used in calculating

Manuscript received April 24, 2012; revised September 5, 2012; accepted October 30, 2012. Date of publication November 8, 2013; date of current version December 20, 2013. This paper was recommended by Associate Editor S. H. Rubin.

The authors are with the Assistive Technologies Research Center, College of Engineering and Computer Science, Wright State University, Dayton, OH 45435 USA (e-mail: Michael.Mills@wright.edu; Nikolaos.Bourbakis@wright.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2012.2227472

maturity [24] and of accuracy, coverage, performance, and scalability; 3) estimated importance of maturity factors to users, developers, and their average scores; and 4) estimated maturities based on the content of the papers and the definitions in Table V. Bar charts include information on the average of estimated accuracy, coverage, performance, and scalability for each method and on maturities of methods that are segmented into regions.

Some methods such as disambiguation impact the effectiveness of other methods. For example, reducing redundancy improves the effectiveness of TE and summaries. NLP or NLU systems may be based on different kinds of methods.

This paper provides functional component analysis and maturity information in an effort to inspire researchers to include more information on maturity, as the research progresses to bridge the gap to where more product information is publically available for graph-based methods in NLP/NLU applications. Finally, the authors have excluded their method to avoid conflict of interest.

II. Methodologies Description

In this section, we summarize a collection of graph-based methods, their performance, and key features that are presented in their corresponding publications. The summary from each paper and a table on maturity definitions were used to estimate maturity numerical scores. Each method number (e.g., method 1) corresponds directly to the reference number (e.g., [1]) of its publication and method numbers (e.g., 1) in various tables and graphs in this paper.

A. Classification by Clustering

Clustering, which was used primarily in methods in [1]–[3] of this survey, provides node classifications, partitions, and pairing or grouping as it traverses a graph representation of the text or group of concepts. Chinese whisper (CW) is an efficient clustering algorithm that works best in parallel or distributed architectures. Other clustering algorithms include finding the medoid of a set of nodes, covering, and finding the centroid of groups of nodes. Clustering is used in these methodologies for the following.

1) Clustering is Used to Induce (or Find) Topically Related Senses From a Graph of Nodes Representing Nouns. Nouns Paired by Edges are Weighted by the Number of Paragraphs, Which Contain Co-occurrences of the Nouns [1]: Method 1 (a through e) represents a group of methodologies that were presented by Korkontzelos *et al*. [1]. In particular, method 1a (for WSI) connects each word with a target word if they co-occur one or more times in context. Graph clustering algorithms are used to induce topically related senses (meanings) of words. Method 1b (for WSI) forms clusters of word pairs. Within a cluster, each vertex stands for a word pair that co-occurs with the target word, and each edge represents the co-occurrence frequency. Method 1c evaluates graph connectivity by using eight measures to estimate free (i.e., tunable) parameters of a word pair. This process is called collocation in graph-based WSI methods. The method identifies the quality of induced clusters (standing for senses) with different parameter combinations, which aid unsupervised parameter estimation in WSI systems. Method 1d uses a corpus of nouns formed from words in context that reference target words. Nouns are removed from the corpus when their distribution within their context falls below a threshold determined by a log-likelihood test. This process removes common (noisy or unimportant) nouns. The result consists of nouns that are more pronounced in determining a sense (or meaning) of target words they reference. Method 1e creates a collocation graph (representing word pairs) with the weight of each collocation calculated from the number of paragraphs, within the corpus, where each word pair exists. A smoothing technique compresses the sparseness of the graph by discovering new edges between vertices. A CW clustering algorithm [2] is used to cluster the collocation graph, thus finding the senses of the target word. Each target word is then tagged with a sense (or meaning) according to the induced cluster [1].

Performance: The eight graph connectivity measures automatically select parameter values that increased performance for both supervised and unsupervised evaluation schemes. Published scores included a recall of 84.8% [1].

Discussion: A log-likelihood test (which finds the maximum of a concave curve) detects the distribution of common nouns that fall below a threshold; therefore, they can be weeded out of a corpus of nouns, leaving more pronounced nouns to determine sense (meaning) of target words. This statistical process has a functionality that is analogous to weeding out unpronounced signals with lower eigenvalues using signal processing filters. Based on other papers in this survey, the CW clustering (or classification) algorithm is an efficient (parallel) algorithm.

2) Clustering Finds Sets of Similar Words and Hierarchies of Concepts by Partitioning Weighed Undirected Graphs [2]: Method 2 [2] is a graph-clustering algorithm, called CW, which partitions nodes into classes of nodes that result in a weighed, undirected graph. The CW algorithm is time linear in complexity to the number of edges in the graph. In document clustering, it finds sets of similar words and concept hierarchies. For WSI applications, the edges represent word co-occurrence with other words in context. The CW algorithm partitions weighted undirected graphs. It finds groups of nodes that broadcast the same message to their neighbors (simulating an agent-based social network) [2].

Performance: Published scores include the following—% recall: for noun 75.5, verb 67.1, and adjective 61.9; % precision: for noun 90.0, verb 77.6, and adjective 92.2 [2].

Discussion: The clustering algorithm turns a graph, which represents a text into a more efficient graph representation that is used for finding (inducing) meanings (senses) of words. Nodes are compressed into clusters of similar words with co-occurrence information contained in the edges. The social network like functionality of the algorithm enables parallel execution for increased performance.

3) Method 3 [3] Summarizes Multidocuments: Clustering Objects Representing Similar Sentences: This method detects the object of the cluster that corresponds to the sentence that most represents the sentences identified within each cluster. Since this methodology uses clusters and objects to detect similar sentences, this was included in this analysis of graph methodologies. Two clustering methods (covering and k-medoid) are used to detect sentences that are similar in content and select the most representative sentence. Covering minimizes the number of clusters (of objects representing sentences) so that the similarity between the most representative sentence object (or medoid) of a cluster and the rest of the objects (sentences) of each cluster exceeds a threshold. K-medoid partitions a set of objects into k clusters so that it maximizes the sum of similarities between the objects and their medoid. A threshold is used in both methods to approximate the minimum or maximum to lessen the exponential complexity of these methods [3].

Performance: Both methods have exponential growth in computational complexity to the number of objects representing sentences from multiple documents. However, the authors state that the algorithms of this methodology are used for redundancy detection in their SUMMA (summarization) system.

Discussion: High complexity may be an issue for this methodology, even by reducing its effect.

B. Methods in [4]–[7] Use Similarity Measure to Measure Similarities of Meaning (i.e., Word Sense) and to Merge Clusters of Similar Concepts

1) Method 4 [4] Uses Machine Learning to Recognize Textual Entailment Using Tree Pairs Representing Syntax in Graph Form: A tree kernel function measures similarity between two trees (one representing a text string and the other representing a hypothesis text string) by counting their common substructures (or fragments). These kernels are applied to graphs that consist of tree pairs. This TE approach emphasizes that kernel methods are needed to manage the large feature space of all possible syntactic tree fragment pairs that are used to form syntactic relations between text and hypothesis in learning algorithms [4].

Performance: This approach was too early in experimental stages for any performance or maturity results. Moschitti and Zanzotto's experiments used support vector machines, which are required resources impacting maturity calculations [4].

Discussion: This is basically a theoretical approach with no real prototype. It reveals the large space complexity from considering all possible syntactic tree fragment pairs.

2) Method 5 [5] Provides Information on Distance Measures Between Nodes in a Graph: Such distances are useful in measuring weighed relationships between vertices in a graph and what they represent such as semantic similarities, intended sense (or meaning), etc. The paper evaluates random walk algorithms and their performance over shortest path algorithms, and describes a new commute time measure. A pseudoinverse of the Laplacian is used to drive estimates for commute times between nodes on a graph. The method uses singular value decomposition (SVD) to discard least significant eigenvectors.

Algorithm description: First, construct a graph, using every pair of words to compute similarity. For each word w in a graph, add an edge between w and all its parts of speech. For each part of speech, add edges to all its senses (or meanings). For each word sense, add edges to all of its hyponyms ("is a") and hypernyms ("includes").

A random walk algorithm captures graph connectivity as well as path lengths and is used to calculate similarity (Pagerank is an example of a random walk algorithm which is used in search applications such as Google) [5].

Performance: This methodology provides a new way to calculate weights of edges based on compute time, which uses the pseudoinverse of the Laplacian represented in a graph and discarding least significant eigenvectors using SVD to reduce noise resulting from the graph construction process. The resulting weights can be used to calculate relatedness or similarity for various types of NLP [5].

Discussion: Distances, in this context, can stand for various important components of NLP including SM, meaning (or sense) induction, etc. The random walk algorithm that was used in this approach and others (within this survey) appears to be an efficient and popular algorithm yielding relatively good performance.

3) Method 6 [6] Uses Weighed, Directed Graphs to Model the Range of Influence of Terms Within a Document and Uses Context to Find the Semantic Relatedness of Terms (Within a Sentence): This is an unsupervised method using no labeled data or training of parameters. A term is represented by a series of nodes in a graph. One node represents a sentence. Weights of edges represent SR (or connection strength) between terms. The method segments topics by finding where cohesion in the graph is weakest, representing the end of a term influence within a document. It uses pointwise mutual information (PMI) to measure similarity (of topics) in nearby sentences. Topical coherence is computed by 1) using the relevance intervals (RI) of each term to model its influence and 2) using other terms in the neighborhood of two connected terms to increase their SR. This reinforcing of related terms measures coherence between sentences. To calculate RI, a 325-million-word corpus (seven years of New York Times) is used with a part-of-speech tagger and a 50-word window to calculate PMI used to measure relatedness between terms. RI is then used to segment a document into each term's range of influence. The result is a weighed graph of connections across slices of the graph, where each slice represents a sentence within a document [6].

Performance: Quotes from the paper related to performance include "... measure of semantic relatedness reinforces global co-occurrence statistics with local contextual information, leading to an improved representation of topical coherence." This text segmentation "models topical coherence using long-range influence of terms and a contextually determined measure of semantic relatedness" [6].

Discussion: This methodology uses weighted, directed graphs to find the range of influence of terms and how they are semantically related.

4) Method 7 [7] Learns Word Semantic Similarity Measures From Traversing a Graph Representation of a Corpus of Parsed Text to Extract Word Synonyms From a Text: The method learns different graph walk models while traversing the graph and uses these models to extract word synonyms. It constrains paths of its random walk to specific word types that match the type of the word of interest. It uses history of its random walk to estimate edge weights to constrain its path, thus pruning paths with probabilities lower than a predetermined threshold [7].

Performance: The authors executed and compared this graph method with two vector methods. Their results showed that "learning specialized graph walk models for different word types" yielded an increase in performance (in accuracy) [7].

Discussion: The learned models that resulted from these random walks, constrained by type information, reduced search space which would likely result in more efficient walks.

C. Methods in [8]–[13] Use Nouns and Verbs, Including Phrases and Modifiers, to Capture Information From a Text and Represent it in Graphs

1) Subjects, Verbs, Objects, and Verb Modifiers Are Used in Graphical Relations to Match Given Questions to Corresponding Answers Derived From the Text [8]: Methods 8a through 8c [8] support finding answers to questions using text entailment. The subject–verb portions of the methodology are implemented as a graphical structure. Method 8a uses a sentence semantic component analysis that encodes dependence trees (produced from the Stanford Parser) that include features used to match components of questions. Method 8b addresses text entailment features from paired sentence analysis. Semantic components (from M8a) of a question are compared with that of each sentence in the text to find an answer from the text. A graph-based semisupervised learning algorithm is used for entailment ranking. Method 8c is a graph summarization algorithm that uses a nearest neighbor approach to estimate representative vertices of a summary dataset. Closer vertices represent similar data points representing denser regions in the hyperspace. A group of similar data points are represented by a new vertex representing all vertices likely to have the same label [8].

Performance: They "demonstrated that summarization on graph-based SSL can improve the QA task performance when more unlabeled data is used to learn the classifier model" [8].

Discussion: Algorithms include a graph-based semisupervised learning algorithm for entailment ranking and a graph summarization algorithm that combines similar data points into single vertices (representing clusters of data).

2) Graph-Matching Algorithm Uses Verbs With Their Arguments and Nouns With Their Modifiers to Build Syntactic–Semantic Graphs. These Verbs and Nouns Come From Sentences, Clauses, and Phrases. Semantic Relations Are Formed From Clauses and Noun Modifiers and Relation Classes, Including Causal, Conjunctive, Participant, Spatial, Temporal, and Quality. The most Important Verb of a Sentence Is Used as a Head Word of the Sentence [9]: Method 9 [9] is a graph-matching algorithm that extracts pairs of syntactic units from a text and assigns a semantic relation to each pair. It incrementally learns from previous pairs and relations it has assigned plus some user feedback. It builds a syntactic–semantic graph as it assigns semantic relations. This methodology matches graphs starting with a small amount of encoded (manually labeled) knowledge from a dictionary. It extracts pairs of text units (such as clauses, a verb and its arguments, and a noun and each of its modifiers). It relies on previously processed examples (stored as graphs) to find the most appropriate relation for an extracted pair from a text. It incrementally learns from semantic analysis [9].

Performance: The authors have stated that "Graph-matching is most useful for assigning semantic relations between verbs and their arguments, but it also gives good results for inter-clause relations." For noun phrases, only noun-modifier pairs with syntactic structure were useful for semantic analysis [9].

Discussion: This algorithm uses subject–verb–object as one of its structures. The purpose of the methodology is for graph matching of the input text with the previously processed text and dictionary markers such as propositions, coordinators, and subordinators associated with semantic relations. Semantic relations for this methodology are highly dependent on which domain is being used. The methodology uses an incremental approach in finding which semantic relations to use between concepts. Accuracy of the semantic representation increases as more of the text is processed.

3) In-Degree Algorithm Uses Nouns, Verbs, Adjectives, and Adverbs to Find the Most Probable Sense (Meaning) of Words Based on Their Context. Hyponyms Found Using Similarity Measures are Included in Determining Word Sense [10]: Method 10 [10] performs WSD using an unsupervised learning approach with both WordNet and SemCor as lexical resources (linked word dictionaries containing synonyms, etc.) and a modification of the graph-based in-degree algorithm to find the most probable sense of each word in a sentence based on context. The modified in-degree algorithm creates a weighed graph with nodes, representing senses of a word and edges with weights connecting the word to each sense. The sum of weights of the edges connecting a node (representing a word) is the in-degree of the node (vertex). The sense with the maximum in-degree value is the selected sense for the word. Similar senses are detected using hypernyms, hyponyms, similar attributes, similar verb groups, pertinyms, holonyms, and meronyms [10].

Performance: The authors state "... outperformed baseline and state of the art using unsupervised system (SM07) in overall accuracy across all data sets" [10]. They modified an in-degree algorithm and used linked dictionary resources: WordNet and SemCor. Published scores at SenseVal (SV) and SemeVal (SM) conferences are as follows: SV2 0.629, SV3 0.603, and SM 0.468 [10].

Discussion: This method performs WSD (i.e., finding the most probable sense of each word based on context) by creating a graph with nodes representing senses of a word and edges with weights connecting the word to each sense.

4) Event Indexing Uses Actors (Based on Current Events), Time of Day (for Present and Previous Events), Causal (Past Event Effects on Current Events), Intention, and Goals of Actors (From Previous Events) to Form Relations Between Concepts. Causal Connections Consist of Subject to Object Relations, Formed From Chains of Verbs and Nouns [11]: Method 11 [11] performs event indexing and summarization using a cognitive psychology approach to creating clusters to build concept representations from a text. Indices include actors in the current event, or times of occurrence of present or previous events, causal relationships of current to previous events, special relations between events, and intention relationships or relationship between goals of actors and the present event. An actor (called protagonist) is a subject that consists of a noun phrase or pronoun referring to a subject or object or a noun phrase in one or more of the previous sentences. (Note that actor or subject is the same as agent. Object is the same as patient.) A causal relationship in this methodology is the relationship of a sentence to previous sentences or in the same sentence. (Note that a causal relationship might include the *action* an agent has on a patient, although not directly referred to in these terms.) Time information is gathered from stated times in a sentence, tenses of verb phrases within a clause, or from the WordNet lexical resource containing temporal relations. Location-related noun phrases are gathered as spatial information within a sentence. The methodology resolves pronouns according to gender and definite pronouns. Causality indexing is used in the noun phrase level by creating causality connections between each pair of subjects and objects, saves the connections, and joins them into chains. The authors use cluster of sentences with indices, cluster filtering, and reduction, and they control the size of the outputted summary [11].

Performance/maturity: The authors describe each algorithm in detail, thus providing a high availability score. Their methodology has been evaluated at the DUC 2003 Conference [11].

Discussion: This event indexing (by actors, times of occurrence and causal, special, and intention relationships) creates clusters to build concepts from a text. Their causality indexing algorithm produces chains of connectivity between subjects, verbs, and objects including their noun and verb synonyms. This approach appears to work best for the text containing lots of events and actions. Descriptive text with little events and actions may not yield causal chains of any significance. However, most applications that need such a capability would find this type of approach useful, assuming that it is mature enough to use.

5) Method by Kozareva [12] Learns Terms That Express Cause–Effect Relations From a Text. It Uses Graph-Based Metrics to Rerank Extracted Information, and Filter and Delete Erroneous Terms to Increase Accuracy of Cause–Effect Relations: This is a bootstrapping method that uses patterns to represent cause–effect relations, learns while recursively building terms of a pattern, reranks what is extracted, and then filters the terms. From a seed term and a recursive pattern as input, the method generates learned terms, which expand the pattern representing a knowledge expansion. The weighed sum of the incoming and outgoing edges of each node provides a ranking which is used with a threshold to delete erroneous terms in the expansion [12].

Performance: The author's evaluation, using SemEval-1 Task-4 which classifies semantic relations between nouns, showed 89% accuracy after ranking 1500 terms.

Discussion: This approach learns cause–effect relations from patterns. The methodology appears intuitively straightforward and yields relatively high accuracy.

6) Method by Liu et al. [13] Learns Event Rules by Identifying Context Dependences From the Parsed Annotated Text. It Extracts Events by Matching Subgraphs of Sentences as It Searches for a Graph of an Event Rule: For event recognition, this method searches for a subgraph that is isomorphic to dependence representations of previously learned event rules. The dependence of event rules is implemented as a labeled directed graph. The union of shortest dependence paths, each representing a training sentence, produces a graph representation of an event [13].

Performance: This method yields 41–52% F-score in detecting and identifying biological events [13].

Discussion: Although isomorphism normally yields a simpler model to work with, searching for an isomorphism is normally a computationally hard problem. Such cases can sometimes be partially resolved with approximate solutions. The authors merge rules across event types to increase precision. They investigated rule ranking to increase accuracy of learned rules.

D. Concept Representations and Their Semantic Relations, Consisting of Graphs and the Methods That Create and Use Them, Are Important Steps Toward Natural Language Understanding. Methods [14], [16], and [17] Include the Concept Level of Representation and Manipulation. Method 15 [15] Serves as an Interim Step Toward This Representation

1) Method [14] Maps Text to Concepts Using Ontology for Concept Extraction from a text and Uses (Biomedical) Domain Knowledge From a Metathesaurus to Find Synonym and "is-a" Relations to Summarize Documents: Clusters are used to identify closely related concepts as a theme in the document. This summarization method includes the following. 1) It represents a document as a graph that consists of concept nodes and relation edges. A unified medical language system (UMLS) metathesaurus is used to identify the correct concept for each term in the text and to disambiguate by extending concepts with their hyponyms (synonyms + is-a relationships). It merges sentence graphs into document graphs. 2) It clusters concepts with similar meaning (or theme) and recognizes themes in document graphs. The most central concepts in a cluster give sufficient and necessary information. 3) Sentences are selected based on similarity between sentences and clusters [14].

Performance: This method to summarize biomedical literature represents a document as an ontology-enriched scale-free graph, the UMLS concepts, and relations. This provides "a richer representation than the one provided by a vector space model."

The authors have "identified several problems and some possible improvements [14]."

Discussion: The use of domain knowledge in this approach provides additional information to guide the mapping from a text to concepts to summarize documents. Intuitively, this added information could aid inference at various nodes in a concept graph to increase resulting summarization accuracies.

2) Method 15 [15] Uses a Random Walk Algorithm Over a Graph Representation While Measuring SR of a Corresponding Text: It provides knowledge integration by using the linked word similarities of Wikipedia to detect similarity of meanings from different words that are found in the text. It computes SR between text pairs by building a graph representation of Wikipedia and using the link structure with different link types as added information for its semantic analysis. Then by using the random walk (or page rank like) algorithm to traverses the graph structure of Wikipedia, it maps each word to the graph creating a randomized vector. The PageRank algorithm computes a stationary distribution for each word in the text from the vectors. A score of each stationary distribution for each word pair provides a measure of vector similarity (or cosine similarity) of the pairs of text (based on all words of the respective text). The authors initialized the Wikipedia graph random walk using two methods: one based on dictionary, and the other based on explicit semantic analysis (ESA). They tested both methods and found ESA to perform better than the dictionary method and slightly better than other published methods using similar approaches with Wikipedia [15].

Performance: The authors state "… even with a simple dictionary-based approach, the graph of Wikipedia links act as an effective resource to compute SR. However, the dictionary approach alone was unable to reach the results of state of the art models using Wikipedia or using the same technique on WordNet." … "by using ESA" (Explicit Semantic Analysis) "to generate the" … "distribution, we were able to introduce small gains using the random walk." … "Performing random walks with personalized PageRank over the Wikipedia graph is a feasible and potentially fruitful means of computing semantic relatedness for words and texts" [15].

Discussion: Wikipedia was used as a dictionary-like knowledge base, due to its enormous capture of knowledge. Other (perhaps more reliable) linked knowledge sources could be used in its place. This conceptual demonstration showed how a graphical representation of SR could be determined and measured.

3) Method 16 [16] Creates a Concept Lattice That Indexes Local Topics Within a Topic Hierarchy. Concepts in the Form of Topics Are Represented by Nodes in the Lattice: This method uses the concept lattice to extract text from multiple documents and generate an optimized summary. The concept lattice provides indexing of local topics within a hierarchy of topics. The topics, which are represented by nodes in the lattice, correspond to concepts that appear frequently throughout the original text or document(s) and sentences that contain these topics. The resulting summary contains an optimized set of local topics and a maximized coverage of concepts for a desired size of summary (number of sentences). This methodology produces a summary by extraction (containing key sentences from the original text) [16].

Performance: Online complexity of using the WordNet lexical dictionary resource to compute all possible senses of each concept is reduced from $O(n\hat{}2)$ to linear by computing sense similarity offline. The method uses a global selection strategy to minimize loss of information from the concept lattice. The methodology was run on document understanding conference (DUC) 2005 and 2006 evaluations [16].

Discussion: This concept lattice approach optimizes coverage of concepts within its graph representation, thus producing optimized summaries from this high level representation. It provides low complexity.

4) Method 17 [17] Constructs Semantic Space Models With Annotated Dependence Relations and Builds Semantic Context From a Dependence Graph, Which Maps Dependence Paths to Words: This semantic representation contains significant linguistic information. The algorithm builds a semantic context of words of interest from dependence paths defined on a dependence graph, specifies dimensions, provides inference over classes of basic elements, and specifies relative importance of different paths [17].

Performance: Experiments in the paper only concentrated on semantic spaces that used a mapping function that maps dependence paths to words. More experimentation is planned [17].

Discussion: This methodology uses syntactic dependences to build semantic space models that contain words of interest in a graphical representation of knowledge. Since this was in the initial experimentation phase, no performance or maturity information is available.

E. Other Methods in [18]–[22] Provide Graph-Based Ranking of Data in Labels, Find Closest Sense of Words in Labels, Detect New Information, Resolve References to Events, and Generate Graphs From Word Co-occurance in a Text

1) Method 18 [18] Uses Unsupervised Learning by Training on a Large Sample of Text (Corpora) to Find the Sense (Meaning) of Words for Labeling Nodes of a Graph: It consists of two graph algorithms for unsupervised induction and tagging of word sense (or intended use of words) based on corpora (sample of text). Agirre *et al.* found that a small sample of nouns was enough to optimize parameters of these algorithms for efficient WSI [18].

Performance: Author's discussion on efficiency is as follows: "Regarding efficiency," their "implementation of HyperLex is extremely fast. Doing the 1800 combinations takes 2 hours in a 2 AMD Opteron processors at 2GHz and 3 Gb RAM. A single run (building the MST – *Minimum Spanning Tree* of the hub of senses connected to target words, mapping and tagging the test sentences) takes only 16 sec" [18].

Discussion: Nouns appear to contain the most helpful information for training algorithms to efficiently determine senses (meanings) of words in a sentence. It also needs more detail on how the two graph algorithms work.

2) Method 19 [19] Detects Novelty (Text, Containing New Information, Compared with what is Already Found) at the Sentence Level by Analyzing Feature Sets from Graph Representations of Sentences: This methodology uses sentence-level term distances (or Kullback–Leibler (KL) divergence, i.e., relative entropy) and PMI to produce graphs that represent sentences.

KL divergence and PMI are calculated as follows.

1) KL divergence $= \Sigma p_d(w) \log [p_d(w)/p_R(w)]$, where w $=$ word shared between document d and document (set) R. p_d and p_r $=$ nonzero probability distributions of words in d and R.
2) $\mathrm{PMI}_{(i,j)} = \log_2[P(i,j)/P(i)P(j)]$, where $P(i,j) =$ number of sentences containing both words w_i and w_j. $P(i)$ and $P(j) =$ number of sentences containing words w_i and w_j, respectively.

The weight of an edge (between vertices I and j) is $\mathrm{wt}_{i,j} = \left[1 + \mathrm{PMI}\,(i,j)\,/d_{ij}^2\right]$ [19].

Features from a term distance (TD) graph are based on strengths and number of connections between words. When novelty (or new information) is detected, the corresponding new sentence graph is added to the existing (background) graph. When the information from the input graph already exists, the weight of the existing edge that represents the information is incremented. The more the graph changes when information from input sentences is added, the more likely the information of the input sentence new. Nineteen graph-based features are used to assess the amount of graph change (i.e., new information) when a new sentence is added. This corresponds to measuring the importance of the new information by detecting the change in scores distributed among the features [19].

Performance: The author states "A highly connected term-distance based graph representation, with the addition of point-wise mutual information, is a computationally relative cheap approach" [19]. Published scores include F-measure of 0.618.

Discussion: KL divergence (also known as relative entropy) is a difference between probability distributions that can be approximated as a distance measure. PMI measures how much one word is associated with another. In this paper, it measures how much certain words occur together in a text. The result provides strengths of connections and number of connections between words in a graph representation.

3) Method 20 [20] Resolves References Within Two or More Sentences to Events: It provides two methods to compute coreference matrices that are used for event coreference resolution in extracting content from a text. (Event coreference consists of references to the same event from two or more sentences, and its resolution resolves or detects classes of these references.) The methodology forms clusters in graphs to represent separate events; therefore, it can resolve (or collect references to the same event) by using the entity-constrained-mention (ECM) F-measure [20].

Performance: The paper investigated event coreference resolution. It used two methods to compute a coreference matrix: 1) computing a coreference formula and 2) applying a maximum entropy model. A high-performance event extractor is required to resolve events. Published scores include F-measures of 0.8363 and 0.8312 [20].

Discussion: Coreference is an important component of NLP since it is used in finding word associations in context. In this case, it is used to find references to events within the text and to improve its accuracy (or resolution).

4) Method 21 [21] Constructs a Word Co-Occurrence Graph for Each Target (Important) Word Using Co-Occurrence Statistics From a Large Set of Text (20 Million Sentences From the New York Times Corpus): Unsupervised learning from the corpus provides graph clusters with features that provide the kind of information normally used in systems with supervised learning. Representative graphs with different parameters are generated from the corpus. The target node in the graph (representing the target word) is directly connected to each node that represents a word significantly co-occurring with the target word in a sentence. The weight for each edge is calculated from the log-likelihood of the co-occurrence of the connected words (derived from the relative frequency gathered from the corpus text). Graph parameters include size (number of co-occurrences or connections) and density. The CW algorithm (see M6) clusters the neighborhood of the graph by partitioning the nodes and representing each cluster as a word sense, thus providing WSI [21].

Performance: The authors state "Co-occurrence cluster system outperforms baseline" (second best in SemEval 2007 lexical sample task). "WSD setup is competitive . . . using minimal linguistic preprocessing and no word sense inventory information . . . except for training examples" [21]. System assigned "acceptable substitutions in over 91% of cases, . . ., with 5.4% error rate" [21]. Published scores include precision of 88.5%.

Discussion: Co-occurring words within a number of sentences within a large corpus of text can transform into a graph and adjust graph-producing parameters. The CW algorithm (also used in M1, M6, and M13) forms clusters from the graph to represent different meanings or word senses, thus providing WSI.

5) Method 22 [22] Uses a Map-Reduced Algorithm in a Semisupervised (Partially Labeled) Graph Representation of

TABLE I
COMPONENT FUNCTIONS

Methodologies:=> [References]	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	
Group of Methods	A,C	A	A,B	B	B	B,D	B	C	C,D	C	A,C	C	C	A,D	D	D	D	E	E	E	E	E	
Authors:=>	Korkontzelos	Biemann	Moens	Moschitti	Rao	Ambwani	Minkov	Celikyilmaz	Nastase	Guo and Diab	Guo and Stylios	Kozareva	Liu	Morales	Yea	Ye	Pado	Agirre	Gamon	Chen	Biemann	Rao	
Capabilities: =>	WSI	WSD, WSI	SM	SM, TE	SM	SR, SM	SM	QA, Sum, TE	SR, Sum	WSD	ER, Sum	ER	ER	WSD	SR	SM	SR	WSD	ND	ER	WSD, WSI	Lab	Total Methods
Functional Components:																							
Causal relationship, connections, indexing											1	1	1										3
Chinese Whisper	1	1																					3
Classification	1	1	1			1													1		1	1	8
Cluster - Concepts - Similar Meaning				1							1				1							1	4
Cluster - Minimize (Partition k clusters)			1																			1	3
Cluster - paired words	1	1																					3
Cluster - Topics																							1
Concept Graph														1									1
Concept Lattice - Topics in lattice											1												2
Concepts (nodes) Relations (edges)				1																			1
Conceptual Knowledge Base																							1
Connectivity of graph	1																				1		3
Co-occurrence	1																	1					3
Co-reference (Find Word Associations)																				1			1
Coverage - Maximize - of Concept			1																				1
Covering (for clustering)																						1	1
Document - Similar, Category, Cluster										1													2
Event - Actors, Time, Causal Relations											1	1											2
Event - Location (Spatial) - Noun-Ph											1												1
Event Indexing											1												1
Events - sentences - time of events.											1	1	1										3
Hubs - (WSD) -																		1					1

Methodologies:=> [References]	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	
Group of Methods	A,C	A	A,B	B	B	B,D	B	C	C,D	C	A,C	C	C	A,D	D	D	D	E	E	E	E	E	
from edges																							
In-Degree Algorithm						1																	1
KL - Divergence									1										1				2
K-Medoid (for clustering)			1																			1	2
Laplacian Inverse																							1
Laplacian Minimum (Linear Equations)																					1		1
Map-reduce Algorithm																					1		1
Mutual Information (also, see PMI)						1													1				2
Nearest Neighbor								1															1
Noun, Verb, Adjective, Adverbs (pairs)						1						1	1										3
Nouns in context	1												1										2
Noun-Verb-Noun phrase												1	1										3
Page Rank Algorithm												1									1		3
Pointwise Mutual Information (PMI)																				1			1
Relative Entropy																			1				1
Random Walk Algorithm									1					1							1		6
Select sentence similar to cluster				1																			1
Semantic Equivalence Recognition						1			1						1		1						5
Sense	1	1		1		1																	6
Similar - Verb groups						1																	1
Similarity		1	1	1		1		1		1										1		1	10
Smoothing - compress sparse	1																						1
Subject-Verb-Object					1			1			1												3
Term distance (sentence level)																			1				1
Text Rank Algorithm -																			1				1
Theme Recognition				1																			1
Thesaurus (to identify concept)				1																			1
Topic - Hierarchy - Index locally																1							1
Topic -related	1		1																	1			4
Verb - Attribute Pair																							1
Verb Dependency Vector													1										2
Weighed Graph						1																	2
Weights - Log Likelihood																							2

TABLE II
MATURITY DEFINITIONS [24]

Acronym	Description
A	Availability - The ability to obtain/implement the system based on the description of the method expressed in mathematical formula, pseudo-code, or compiled code. A higher score indicates that a satisfactory amount of information is presented in the description to replicate the system. For example, a system with a score of 10 will contain a clear description of the method and code that could be implemented; whereas a system with a score of 5 may only have a mathematical formula and short process description.
Co	Cost - The amount of money needed to use and/or implement the system based on the description provided. This score reflects the cost of equipment as well as implementation complexity.
FI	Further Improvements - The methodology has the potential for further enhancement. A higher score indicates that a methodology can be improved upon, whereas a system with a lower score is considered more mature and less likely to be improved upon.
MC	Model Complexity - Complexity of model used in the methodology. For example a system utilizing a neural network or wavelet is considered more complex than one that uses a run length smoothing algorithm.
O	Originality - The methodology is based on original algorithms and/or mathematical operations; or the synergistic combination of simple methods composing a new method. A method that is referenced in the literature as original is given a higher score than one that builds on another method.
P	Prototype - The methodology has been successfully implemented at the experimental stage and produced desirable results. Scores for this aspect were also affected by the results presented. A paper that presented comparative results scored higher than one that presents an illustrative example.
RP	Released Product - The methodology has been implemented in a commercial setting. This aspect has a value of either 1 or 3, where the few methods/systems that have been utilized in a commercial setting are given a slight advantage over others.
Re	Reliability - The methodology produces expected results under normal operating conditions.
Ro	Robust - The methodology produces acceptable results under extenuating circumstances. This score is based on the features of the methodology as compared to methodologies in a similar category.
Sp	Speed - Reported processing time for sample tests. Note that some authors do not report performance metrics. For these we give a score of 3 out of 10.
U	Usability - The methodology offers a user-friendly interface so that the user can work easily with it. Systems that require no user input are given a higher score than those that require input parameters or training data.
M	Maturity - A measure that combines the scores of the different aspects. Maturity = U + O + ((A*P*RP) + (Re*Ro*Sp))/(Co*FI*MC)

TABLE III
ACCURACY, COVERAGE, PERFORMANCE, AND SCALABILITY VALUES

Acc	Accuracy – Captures intended meaning (sense) at concept and lexical representations and generated text.
Cov	Coverage – Handles nearly all senses of a word. Disambiguates all (at least, important) words. Captures all if not most of meaning in generated summary or text entailment.
Per	Performance – Resulting scores in precision, recall, and f-measure from testing Conferences such as Document Understanding Conferences (DUC), Disambiguation Evaluation (SensEval), and Semantic Evaluation (SemEval). Values are normalized to a compatible scale with other factors in the table.
Sca	Scalability – Handles multiple and large documents. Handles large database or uses large dictionary.
App	Application: **Ent** (Entailment), **ER** (Event Resolution), **GrA** (Grammar Annotation), **IM** (Information Mining), **K** (Knowledgebase), **Lab** (Labeling), **ND** (Novelty Detection), **QA** (Question/Answer), **Red** (Redundancy Reduction), **Sim** (Similarity Measure), **SR** (Semantic Relatedness), **Sum** (Summarization), and **WSD** or **WSI** (Word Sense Disambiguation or Induction).

TABLE IV
DEVELOPER AND USER WEIGHTS

Method	A	Co	FI	MC	O	P	RP	Re	Ro	Sp	U	Acc	Cov	Per	Sca
Developer	10	10	7	10	8	10	10	10	10	9	9	10	10	10	10
User	7	10	3	2	1	6	10	10	10	10	10	10	10	10	10
Average Weight/10	0.85	1	0.5	0.6	0.5	0.8	1	1	1	1	1	1	1	1	1

Natural Language Text for Label Propagation: The authors have modified the algorithm to be scalable for very large graph representations with linear processing time and localized in memory space; therefore, it is suitable for parallel processing, making its memory space also scalable. They show the importance of label propagation by providing a scalable parallel algorithm that can provide ranking of nodes to derive lexical relatedness between terms applied to disambiguation, paraphrasing, question answering, and machine translation. It also provides polarity induction for sentiment mining applications. The graph-based map-reduced algorithm is adapted for large scalable graph representations of natural language text and applied to label propagation using a semisupervised (with nodes initially 20% labeled and 80% unlabeled) classification approach. Label propagation is a random walk methodology. The algorithm provides linear computation and localized space for efficient memory allocation/deallocation and captures connectivity structure besides distance measure. Label propagation uses the small percentage of labeled nodes in a large graph to define the probability over the labels (including calculating probability distribution among unlabeled nodes). Random walks over a weighed undirected graph calculate minimum of the Laplacian by solving a system of linear equations (instead of the computational complexity of using dense matrices that represent pseudoinverses of the Laplacian), thus arriving at a distance measure of the graph with linear and parallelized computations. The authors associate label propagation with the popular PageRank random walk model which can quickly jump to its initial state, thus saving computation time. The algorithm is parallelized so that it can use local information, thus using constant memory and reducing the required data access to contiguous chunks of memory. This significantly reduces data accesses (number of memory read/writes) [22].

Performance: Random walks for classification involve constructing the graph Laplacian and using its pseudoinverse as a kernel.However, for very large graphs, the pseudoinverses are dense matrices *requiring* $O(n^2)$ *space* which can be *prohibitive*. Author's alternative: Their iterative label propagation algorithm can parallelize using a map-reduced model (introduced by Dean

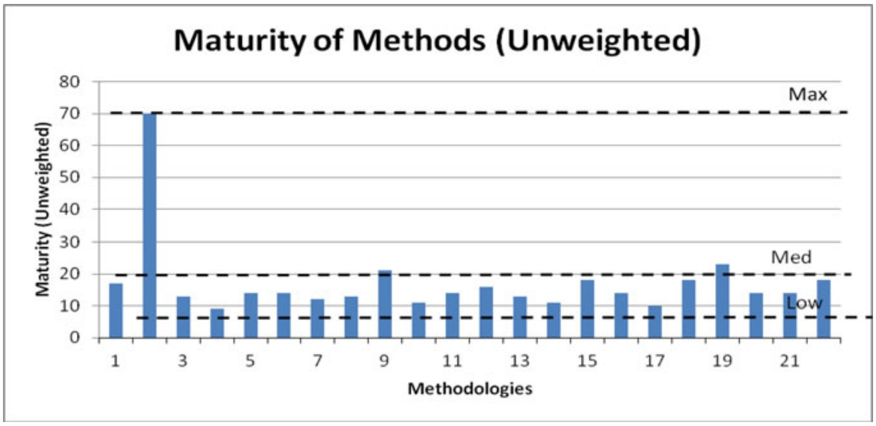

Fig. 1. Maturity of methods (unweighted).

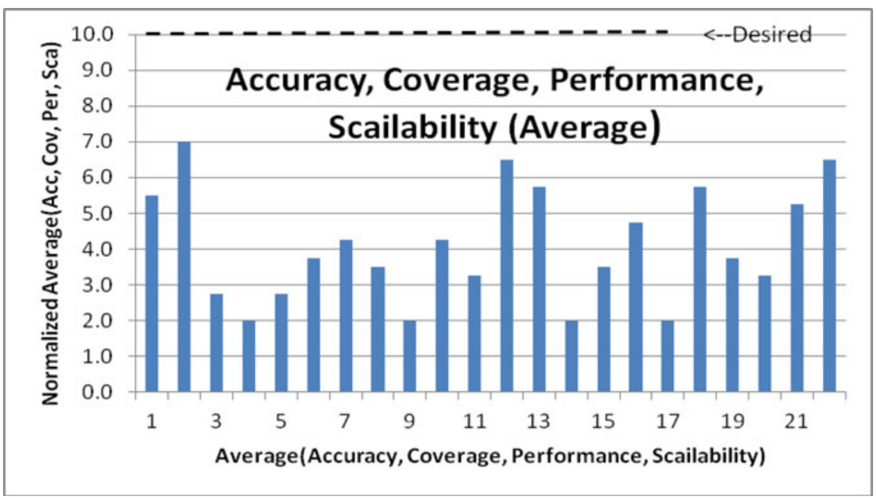

Fig. 2. Accuracy, coverage, performance, scalability (average).

TABLE V
MATURITY OF METHODS (ESTIMATED, UNWEIGHTED) PLUS THEIR ACCURACY, COVERAGE PERFORMANCE, AND CAPACITY

	Maturity												Other Factors			
Method	A	Co	FI	MC	O	P	RP	Re	Ro	Sp	U	M	Acc	Cov	Per	Sca
1	10	1	3	5	4	8	1	7	3	3	3	17	7	6	5	4
2	10	1	3	3	5	8	1	5	8	7	5	70	5	5	9	9
3	6	5	3	8	7	9	2	3	5	2	3	13	3	3	2	3
4	10	5	5	7	7	5	1	2	2	2	2	9	2	2	2	2
5	10	1	3	8	9	3	1	2	2	5	3	14	3	4	2	2
6	9	1	7	8	9	9	1	3	3	3	3	14	4	3	3	5
7	7	3	5	6	7	8	1	5	2	6	4	12	6	4	5	2
8	10	1	5	8	8	8	1	4	3	3	2	13	5	4	2	3
9	9	1	4	7	9	7	3	3	3	3	4	21	2	2	2	2
10	9	1	7	6	5	8	1	2	4	6	3	11	4	4	6	3
11	9	2	6	5	9	5	1	3	5	4	3	14	3	3	4	3
12	8	3	7	6	8	8	1	5	5	6	6	16	8	4	6	8
13	6	3	7	8	8	7	1	5	4	4	4	13	6	6	4	7
14	9	2	8	5	9	1	1	2	2	2	2	11	2	2	2	2
15	10	1	7	5	7	8	1	3	4	4	7	18	2	4	5	3
16	8	2	3	3	4	8	1	3	4	3	4	14	4	5	7	3
17	3	2	9	5	9	3	1	1	1	1	1	10	2	2	2	2
18	10	1	3	7	4	8	1	5	3	8	4	18	8	5	6	4
19	9	1	3	5	9	8	1	4	3	6	4	23	4	4	4	3
20	10	1	5	7	9	8	1	3	2	3	2	14	3	2	6	2
21	6	1	3	7	6	8	1	3	3	3	4	14	9	3	3	6
22	10	1	4	9	9	8	1	3	8	7	2	18	5	5	7	9

and Ghemawat) [23] with its solution derived from a set of linear equations. The algorithm uses fixed memory regardless of the size of the graph and *scales linearly* in the size of data and the number of processing elements in the cluster. Published scores are as follows: F-measures for nouns 0.5853, verbs 0.8340, and adjectives 0.7295 [22].

Discussion: The methodology ranks nodes to derive lexical relatedness between terms which makes it useful for disambiguation, paraphrasing, question answering, and machine translation. The label propagation it produces transforms a partially labeled graph into a more fully labeled graph which could be interpreted as a semisupervised approach in automatically applying word sense labels for disambiguation. The treatment of scalability issues for large data applications by scaling graphs linearly reduces space complexity significantly.

III. FUNCTIONAL COMPONENTS OF GRAPH-BASED METHODOLOGIES

Table I dissects the information from each method into its functional components and method capabilities to help present a functional analysis encompassing all NLP and NLU methods that are summarized in this survey. It shows functions, capabilities, names of algorithms, and type of learning for each methodology. Major functions of interest from this table are discussed within the conclusions. Capabilities (abbreviated in Table I and described in Section I) include ER, GrA, IM, K,

TABLE VI
MATURITY SCORES OF METHODS (WEIGHT × ESTIMATED MATURITY)

	Maturity												
Method*	A	Co	FI	MC	O	P	RP	Re	Ro	Sp	U	M	**M normalized**
1	8.5	1.0	1.5	4.0	1.8	6.4	1.0	6.3	2.4	2.9	2.6	21	2.7
2	8.5	1.0	1.5	2.4	2.3	6.4	1.0	4.5	6.4	6.7	4.3	75	10.0
3	5.1	5.0	1.5	6.4	3.2	7.2	2.0	2.7	4.0	1.9	2.6	8	1.0
4	8.5	5.0	2.5	5.6	3.2	4.0	1.0	1.8	1.6	1.9	1.7	5	0.7
5	8.5	1.0	1.5	6.4	4.1	2.4	1.0	1.8	1.6	4.8	2.6	10	1.4
6	7.7	1.0	3.5	6.4	4.1	7.2	1.0	2.7	2.4	2.9	2.6	10	1.3
7	7.0	3.0	5.0	6.0	7.0	8.0	1.0	5.0	2.0	6.0	4.0	12	1.6
8	10.0	1.0	5.0	8.0	8.0	8.0	1.0	4.0	3.0	3.0	2.0	13	1.7
9	6.0	3.0	2.5	4.8	3.2	6.4	1.0	4.5	1.6	5.7	3.4	9	1.2
10	7.7	1.0	3.5	4.8	2.3	6.4	1.0	1.8	3.2	5.7	2.6	10	1.3
11	7.7	2.0	3.0	4.0	4.1	4.0	1.0	2.7	4.0	3.8	2.6	10	1.3
12	6.8	3.0	3.5	4.8	3.6	6.4	1.0	4.5	4.0	5.7	5.1	12	1.5
13	5.1	3.0	3.5	6.4	3.6	5.6	1.0	4.5	3.2	3.8	3.4	8	1.1
14	7.7	2.0	4.0	4.0	4.1	0.8	1.0	1.8	1.6	1.9	1.7	6	0.8
15	8.5	1.0	3.5	4.0	3.2	6.4	1.0	2.7	3.2	3.8	6.0	15	2.0
16	7.7	2.0	4.0	4.0	4.1	0.8	1.0	1.8	1.6	1.9	1.7	6	0.8
17	2.6	2.0	4.5	4.0	4.1	2.4	1.0	0.9	0.8	1.0	0.9	5	0.7
18	8.5	1.0	1.5	5.6	1.8	6.4	1.0	4.5	2.4	7.6	3.4	21	2.9
19	7.7	1.0	1.5	4.0	4.1	6.4	1.0	3.6	2.4	5.7	3.4	24	3.2
20	8.5	1.0	2.5	5.6	4.1	6.4	1.0	2.7	1.6	2.9	1.7	11	1.4
21	5.1	1.0	1.5	5.6	2.7	6.4	1.0	2.7	2.4	2.9	3.4	12	1.6
22	8.5	1.0	2.0	7.2	4.1	6.4	1.0	2.7	6.4	6.7	1.7	18	2.3

Lab, ND, QA, Red, SR, SM, Sum, TE, WSD, and WSI. The 85 individual functional components that are collected from all the methodologies discussed in this survey and listed in Table I can be categorized into the functional areas. Major functional areas of interest include causality (including noun–verb–noun relations), clusters, concept representations (including semantic representations), events, sense (or meaning) determination, and distance measure (of similarities, terms, meanings, etc.).

IV. DEFINITIONS

Table II [24] shows definitions for each maturity element used for calculating maturity. Numerical values for each maturity element are extracted from information in technical papers describing each method. The purpose of each maturity table is to show the relative state of the research area as a whole. Included in the table of weighed maturity features are estimated accuracy, coverage, performance, and scalability numbers from both developer and user perspectives. The tables and charts in this survey contain estimated scores that are based on information available in this area and are not intended to reveal how any one methodology is better than another.

V. MATURITY TABLES AND CHARTS

Table IV provides estimated importance factors for developers and users. These factors are used to weigh maturity and other (accuracy, coverage, performance, and scalability) scores in terms of their average importance. What is important to developers (such as mathematical descriptions or pseudocode) is sometimes different from what is important to users (such as user friendliness) and, therefore, need to be weighted differently. However, both developers and users want the maximum accuracy, coverage, performance, and scalability since they show favorable results from developers' efforts and help users achieve their requirements as well.

Table V contains an estimated maturity of each method, which is calculated from maturity components that are defined in

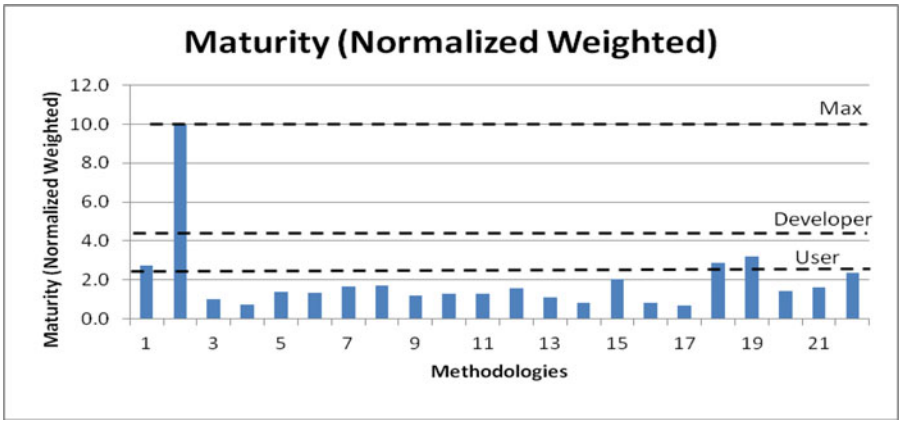

Fig. 3. Maturity (normalized weighted).

Table II. Table V also includes estimated accuracy, coverage, performance, and scalability values with the column headings defined in Table III. The acronyms in the column headings are also defined in Tables II and III.

The following bar charts (see Figs. 1 and 2) show plots from values of Table V containing maturity components, resulting maturities, and other (accuracy, coverage, performance, and scalability) values. Fig. 1 shows maximum, medium, and low lines of reference.

Fig. 2 shows the desired and average lines of reference. Since desired values of both developer and user were at the maximum (best possible value), all of the methodologies fell significantly below the desired score for accuracy, coverage, performance, or scalability values.

Table VI contains weighted maturity component, weighted maturity, and normalized maturity scores. The scores in Table V are multiplied by the average of the developer and user levels of maturity importance from Table IV. Thus, Table VI is weighted toward developer and user points of view. The bar graph in Fig. 3 contains maturity scores after they are weighted by user and developer importance factors and normalized to a maximum of 10. Reference lines in Fig. 3 show estimated, minimum, desired scores of developers and users and the maximum score, all normalized.

VI. Conclusion

The conclusions to this survey have divided into two portions. Section VI-A discusses major functional components that are selected from methodologies summarized in this survey and categorized in Table I. Section VI-B provides maturity conclusions that are based on the corresponding tables and graphs in this survey. It also includes accuracy, coverage, performance, and scalability conclusions.

A. Major Functional Components

The organization of major functional components in Table I shows what NLP and NLU capabilities can be produced using certain functional components existing in different collections of methods from this survey. The conclusions for methods are organized into the following groups: 1) clustering; 2) similarity measure; 3) influence of noun, verbs, and their modifiers, 4) concept and semantic representations; and 5) other methods that do not belong to a particular group are listed later.

1) Clustering: Clustering can filter out common (relatively unimportant) nouns from a graph, leaving more pronounce nouns to determine the sense of words. The result is a more efficient, compressed graph representation of text. It can also calculate similarity measures to detect similar documents and reduce redundancy at the concept level.

2) Similarity Measure: Distances in graphs (such as number of nodes between start and finish) can represent important components of NLP including similarity and senses (i.e., meaning) of words. This is normally accomplished while traversing nodes using an algorithm such as random walk. Weighed, directed graphs can be used to find the range of influence of terms, and how they are semantically related.

3) Nouns and Verbs Plus Their Phrases and Modifiers:

1) Within a corpus used for training, nouns were found to have a higher discriminative ability than verbs, adverbs, and adjectives.
2) Subjects, verbs, objects, and verb modifiers that are used in graphical relations enable entailment ranking and graph summarization algorithms to combine similar data points into single vertices to represent clusters of data points, which compress the graph representation.
3) An in-degree algorithm uses nouns, verbs, adjectives, and adverbs to find the most probable sense of words based on context by creating a weighed graph with nodes representing senses of a word and edges with weights connecting the word to each sense.
4) An event indexing method produces chains of connectivity between subjects, verbs, and objects including their noun and verb synonyms. This approach appears to work best for the text containing lots of events and actions.

4) Concept Representations Are Created in the Following Ways:

1) Use domain knowledge with ontology to map text to concepts. This provides additional information to summarize

documents. Intuitively, this added information would add inference at various nodes in a concept graph to increase the resulting summarization accuracies.

2) Create a concept lattice to index local topics within a topic hierarchy. This optimizes coverage of concepts within its graph representation which optimizes summaries.

3) Build semantic space models from syntactic dependences to produce a graphical representation of knowledge.

5) Other Methods not Fitting in a Group: Provide graph-based ranking of data in labels, find closest sense of words in labels, detect new information, generate grammars for a given text, resolve references to events, and generate graphs from word co-occurrence in a text.

B. Maturity

The following maturity discussion tells how some maturity values were derived from some respective papers. Methods with relatively high maturity values such as method 2 resulted because of their use of well-known and used algorithms such as CW clustering algorithm which is used by major search engines. However, originality, in this case, was low because of the use of algorithms that were already developed earlier by others. The factors that were multiplicative rather than additive in the maturity calculation had a larger impact on the final values. Since originality was additive, its impact on maturity was significantly less than the multiplicative factors such as robustness. Collective results of the 22 methods averaged to a relatively low value. Thus, more work needs to be done on maturing NLP and NLU research so that results can better meet developer and user needs. Results of accuracy, coverage, performance, and scalability showed that the average as well as the maximum scores still left significant room for improvement to meet desires of both developers and users.

1) This Survey Has Provided a Summarization of Graphical NLP and NLU Methods (Labeled as 1 Through 22) With an Analysis of Estimated Accuracy, Coverage, Performance, Scalability, and Estimated Maturity Based on the Information Available in the referenced papers: The tables of values show that this area of research, on the average, needs further improvement before they meet the desires of developers and users of products from this research area. Although significant improvements have been made over the years, a majority of these methods are still in progress. Some of the publications reviewed for this survey showed that systems have been developed from this research area and are used mostly in the medical field, but more mature systems are needed. Very little on maturity has been reported in these research areas. Maturity values that were calculated in this survey were estimated based on information provided in the papers. As this research area continues to progress, more efforts and published results on maturity should help system developers and users meet their needs in various NLP and NLU applications.

REFERENCES

[1] I. Korkontzelos, I. Klapaftis, and S. Manandhar, "Graph connectivity measures for unsupervised parameter tuning of graph-based sense induction systems," in *Proc. NAACL HLT Workshop Unsupervised Minimally Supervised Learning Lexical Semantics*, Jun. 2009, pp. 36–44.

[2] C. Biemann, "Chinese whispers—An efficient graph clustering algorithm and its application to natural language processing problems," in *Proc. Workshop TextGraphs, HLT-NAACL*, Jun. 2006, pp. 73–80.

[3] M. F. Moens, R. Angheluta, and J. Dumortier, "Generic technologies for single- and multi-documents summarization," *Inf. Process. Manage.*, vol. 41, pp. 569–586, 2005.

[4] A. Moschitti and F. M. Zanzotto, "Encoding tree pair-based graphs in learning algorithms: The textual entailment recognition case," in *Proc. 3rd Textgraphs Workshop Graph-Based Algorithms Natural Language Process.*, Aug. 2008, pp. 25–32.

[5] D. Rao, D. Yarowsky, and C. Callison-Burch, "Affinity measures based on the graph Laplacian," in *Proc. 3rd Textgraphs Workshop Graph-Based Algorithms Natural Language Process.*, Aug. 2008, pp. 41–48.

[6] G. Ambwani and A. R. Davis, "Contextually-mediated semantic similarity graphs for topic segmentation," in *Proc. Workshop Graph-based Methods Natural Language Process.*, Jul. 2010, pp. 60–68.

[7] E. Minkov and W. W. Cohen, "Graph based similarity measures for synonym extraction from parsed text," in *Proc. TextGraphs-7 Workshop ACL*, Jul. 2012, pp. 20–24.

[8] A. Celikyilmaz, M. Thint, and Z. Huang, "A graph-based semi-supervised learning for question-answering," in *Proc. Joint Conf. 47th Annu. Meeting ACL and 4th Int. Joint Conf. Natural Language Processing of AFNLP*, Aug. 2009, pp. 719–727.

[9] V. Nastase and S. Szpakowicz, "Matching syntactic-semantic graphs for semantic relation assignment," in *Proc. Workshop Graph Based Method for Natural Language Processing*, Jun. 2006, pp. 81–88.

[10] W. Guo and M. T. Diab, "Improvements to monolingual English word sense disambiguation," in *Proc. Workshop Semantic Evaluations: Recent Achievements Future Directions*, Jun. 2009, pp. 64–69.

[11] Yi Guo and G. Stylios, "An intelligent summarization system based on cognitive psychology," *Inf. Sci.*, vol. 174, pp. 1–36, 2005.

[12] Z. Kozareva, "Cause–effect relation learning," in *Proc. TextGraphs-7 Workshop ACL*, Jul. 2012, pp. 39–43.

[13] H. Liu, R. Komandur, and K. Verspoor, "From graphs to events: A subgraph matching approach for information extraction from biomedical text," in *Proc. BioNLP Shared Task 2011 Workshop*, Jun. 2011, pp. 164–172.

[14] L. P. Morales, A. D. Esteban, and P. Gervás, "Concept-graph based biomedical automatic summarization using ontologies," in *Proc. Textgraphs-3: Workshop Graph-Based Algorithms Natural Language Process. Cooling 2008*, Aug. 2008, pp. 53–56.

[15] E. Yea, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa, "WikiWalk: Random walks on Wikipedia for semantic relatedness," in *Proc. 4th Textgraphs Workshop Graph-Based Algorithms Natural Language Process.*, 2009, pp. 41–49.

[16] Ye Shiren, T-S Chua, M-Y Kan, and L. Qiu, "Document concept lattice for text understanding and summarization," *Inf. Process. Manage.*, vol. 43, pp. 1643–1662, 2007.

[17] S. Pado and M. Lapata, "Dependency-based construction of semantic space models," *Assoc. Comput. Linguist.*, vol. 33, no. 2, pp. 161–199, 2007.

[18] E. Agirre, D. Martínez, O. López de Lacalle, and A. Soroa, "Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm," in *Proc. Workshop TextGraphs, HLT-NAACL*, New York, Jun. 2006, pp. 89–96.

[19] M. Gamon, "Graph-based text representation for novelty detection," in *Proc. Workshop TextGraphs, HLT-NAACL*, Jun. 2006, pp. 17–24.

[20] Z. Chen and H. Ji, "Graph-based event coreference resolution," in *Proc. Workshop Graph-Based Methods Natural Language Process.*, Aug. 2009, pp. 54–57.

[21] C. Biemann, "Co-occurrence cluster features for lexical substitutions in context," in *Proc. Workshop Graph-Based Methods Natural Language Process.*, Jul. 2010, pp. 55–59.

[22] D. Rao and D. Yarowsky, "Ranking and Semi-supervised classification on large scale graphs using map-reduce," in *Proc. Workshop Graph-Based Methods Natural Language Process.*, Aug. 2009, pp. 58–65.

[23] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. 6th Symp. Operating Systems Design Implementation*, 2004, pp. 137–149.

[24] R. Keefer and N. Bourbakis, "Interactions with a mobile reading device for the visually impaired," in *Proc. 21st Int. Conf. Tools with Artif. Intel.*, Nov. 2009, Newark, NJ, pp. 229–236.

Michael T. Mills is currently working toward the Ph.D. degree at Assistive Technologies Research Center, Wright State University, Dayton, OH, where he is involved in conducting natural language understanding research.

He managed the research and development of Air Force systems and lab research programs for 37 years. He was a Technical Reviewer for the development and international standardization of programming languages including Ada95 and a Project Engineer for software tool developments.

Nikolaos G. Bourbakis (F'96) received the B.S. degree in mathematics from the National University of Athens, Athens, Greece, and the Ph.D. degree in computer science and computer engineering from the Department of Computer Engineering and Informatics, University of Patras, Patras, Greece, in 1983.

He is a Distinguished Professor in Informatics & Technology and the Director of the Assistive Technologies Research Center, Wright State University, OH. His previous academic positions are as follows: Associate Dean for Research, Associate Director of the Center on Intelligent Systems, Director of the Image-Video-Vision & Applied AI Research Lab, Professor of Electrical Engineering with joint appointment to the Computer Science Department at SUNY-Binghamton, Professor and Lab Director at Technical University of Crete, Greece, and Assistant Professor at GMU. His industrial experience includes service to IBM, CA, and Soft Sight, NY. He is the Founder and Vice President of the AIIS, Inc., NY. He pursues research in assistive technologies, applied AI, machine vision, biometrics, bioengineering, information security, and parallel/distributed processing funded by the U.S. and European government and industry.