




## Article

# Text Semantic Annotation: A Distributed Methodology Based on Community Coherence

Christos Makris , Georgios Pispirigos  and Michael Angelos Simos 

Department of Computer Engineering and Informatics, University of Patras, 26504 Patras, Greece

\* Correspondence: makri@ceid.upatras.gr (C.M.); pispirig@ceid.upatras.gr (G.P.);  
asimos@ceid.upatras.gr (M.A.S.); Tel.: +30-2610-996-968 (C.M.)

Received: 23 May 2020; Accepted: 25 June 2020; Published: 1 July 2020



**Abstract:** Text annotation is the process of identifying the sense of a textual segment within a given context to a corresponding entity on a concept ontology. As the bag of words paradigm's limitations become increasingly discernible in modern applications, several information retrieval and artificial intelligence tasks are shifting to semantic representations for addressing the inherent natural language polysemy and homonymy challenges. With extensive application in a broad range of scientific fields, such as digital marketing, bioinformatics, chemical engineering, neuroscience, and social sciences, community detection has attracted great scientific interest. Focusing on linguistics, by aiming to identify groups of densely interconnected subgroups of semantic ontologies, community detection application has proven beneficial in terms of disambiguation improvement and ontology enhancement. In this paper we introduce a novel distributed supervised knowledge-based methodology employing community detection algorithms for text annotation with Wikipedia Entities, establishing the unprecedented concept of community Coherence as a metric for local contextual coherence compatibility. Our experimental evaluation revealed that deeper inference of relatedness and local entity community coherence in the Wikipedia graph bears substantial improvements overall via a focus on accuracy amelioration of less common annotations. The proposed methodology is propitious for wider adoption, attaining robust disambiguation performance.

**Keywords:** text annotation; word sense disambiguation; ontologies; Wikification; community detection; Louvain algorithm; Clauset-Newman-Moore algorithm

## 1. Introduction

Word Sense disambiguation is the process of identifying the sense of a textual segment of a sentence. The task deterministically yields a unique mapping to an entity, usually drawn from a concept ontology. Semantic ambiguity is a common property of natural language corpora, hence superficial approaches like the classic bag of words paradigm is often proven insufficient. To that end, word sense disambiguation and text annotation is a common pre-processing step of several information retrieval and natural language processing tasks, such as machine-translation, text summarization, natural language inference, taxonomy learning, searching, clustering, classification, argumentation mining, and question answering, with a major impact on several industries, including artificial intelligence, advertising, and the Internet.

Many recent approaches focus on employing deep neural network architectures, requiring substantial reduction of the training input dimensionality due to their computational challenges for large-scale datasets. However, disambiguation accuracy is largely incommensurate in this computational complexity tradeoff. Statistical dependencies are projected to a lower degree space at the consequence of input accuracy loss. In addition, well-established features in domain knowledge are neglected and delegated for inference during the training phase. The high computational requirements

and lack of interpretability of deep neural network approaches have been inhibiting factors for greater adoption.

In recent years, graph analysis techniques have been broadly adopted due to their supreme performance and their inherent efficiency in handling hierarchical data. From sociology, chemistry, and biology, to finance and computer science, the application of graph processing methods has proven beneficial in revealing the dynamics and the implicit correlations between information entities.

As regards to recommendation systems, social influence analysis, digital marketing, etc., community detection [1] has proven essential in partitioning the original data graph to densely intra-connected subgroups of highly similar entities—that is, communities—by solely leveraging the network topology. Specifically, by recursively calculating a repetitively modified set of network topology metrics such as the edge/node connectivity, the clustering coefficient, the edge betweenness, and the centrality, the community detection algorithms manage to accurately reveal the underlying community structure for any possible information network.

In this paper, a novel distributed methodology based on the community detection of semantic ontologies is introduced in terms of efficient word-sense disambiguation. In particular, by following the divide-and-conquer paradigm, this approach focuses on partitioning the sources of knowledge, expressed as an information network, using a community detection algorithm to identify the densely intra-connected communities of semantic ontologies.

Thorough experimentation has been conducted on Wikipedia's source of knowledge using either the Louvain [2] or the Clauset–Newman–Moore [3] community detection algorithms, which are both considered golden standards in revealing the subjacent community structure. As practically shown, the accuracy and the relevance of the extracted documents are unquestionably ameliorated by focusing the input queries to a finer and far more precise subset of semantic ontologies.

The remainder of this manuscript is organized as follows. In Section 2 the necessary background is presented. In Section 3 the proposed methodology and its implementation are comprehensively analyzed. In Section 4 the experimentation process is described and assessed. Finally, in Section 5 the final conclusions accompanied by some potential improvements are outlined.

## 2. Background

Wikipedia is a crowdsourcing online encyclopedia with millions of articles, constituting one of the largest online open repositories of general knowledge. The articles of Wikipedia are created and edited by a large and highly active community of editors, converging diverse knowledge to widespread and commonly accepted textual descriptions. Each article can be interpreted as a distinct knowledge entity.

Word Sense Disambiguation (WSD) heavily relies on knowledge at its very core. In fact, it would be impossible for both humans and machines to identify the appropriate sense of a polysemous mention within a context without any kind of knowledge. As the manual creation of knowledge resources is an expensive and time-consuming effort, posing copious challenges as new domains and concepts arise or change over time, the matter of knowledge acquisition has been outlined as a prevalent problem in the field of Word Sense Disambiguation.

Several manually maintained ontologies have been employed in the WSD task, outlining the maintainability challenges in the scope of knowledge acquisition. To that end, several approaches have emerged for unifying several knowledge bases yet facing accuracy challenges in the task. As a result, Entity Linking with Wikipedia is one of the most popular recent approaches to the Word Sense Disambiguation domain employed in several similar works [4–23], as Wikipedia's online nature inherits the main principles of the web in a wide and highly active user base.

### 2.1. Word Sense Disambiguation Algorithms

Word Sense Disambiguation and Entity Linking (EL) tasks have been acknowledged as some of the most challenging in the research field of Natural Language processing, due to a variety of factors. The Word Sense Disambiguation (WSD) task has been described as an AI-complete problem [4,5],

namely a problem with equivalent difficulty to that of central artificial intelligence (AI) problems, by analogy to NP-completeness in complexity theory. The task has been approached using a variety of formalizations, at granularities ranging from ontologies and sense inventories to domains, with the presence of domain restrictions or not. The disambiguation coverage focal points, i.e., one target sense per sentence versus full-sentence disambiguation, varies. As WSD relies on knowledge for acquiring the means to associate the most appropriate sense representation in a given context, the challenge of knowledge acquisition, aka the knowledge acquisition bottleneck [6], becomes prevalent.

Word Sense Disambiguation approaches can be distinguished into two main classes according to [5]:

- Supervised WSD: leveraging machine-learning techniques for training a classifier from labeled input training sets, containing appropriate sense label mappings along with other features.
- Unsupervised WSD: methods based on unlabeled corpora, lacking manual sense-tagged input context.

Another distinction comprises the use of external lexical resources, such as ontologies and machine-readable dictionaries, with knowledge-based, aka knowledge-rich, and corpus-based, aka knowledge-poor, approaches, respectively. Several recent research endeavors have focused on Supervised Knowledge-Based WSD methodologies, using Wikipedia as the underlying knowledge ontology for the entity linking task.

The first works on the Word Sense Disambiguation problem following the entity linking approach were [7,8], which proposed suites of novel methodologies for text annotation with hyperlinks to Wikipedia articles and presenting the technical aspects and challenges of the task.

In [9,10], some more advanced methods were proposed introducing relatedness between Wikipedia articles as the overlap of their inbound links along with the concept of coherence of a candidate annotation with other unambiguous mentions within a context, while modeling the disambiguation process as the selection of the annotations that maximize a global score ranking of coherence relatedness and other statistics.

In [11], further improvements were achieved by segmenting the ranking scores to local, for modeling context information concerning only the specific annotation, and global, for modeling the sense consensus coherence among all of the annotation link selections in a text and their disambiguation. The selection ranking was formalized as a quadratic assignment problem for approximating the entity selection maximizing the sum of global and local scores.

Later, [12] introduced a computationally efficient and accurate local disambiguation methodology with a focus on short texts that aims at disambiguating using a voting scheme for ranking candidate entity annotations, and selecting the top-scoring one using the collective agreement between all the annotations within a given context, by employing relatedness, commonness, and other statistics, derived by Wikipedia export pre-processing.

The authors of [13] explored an alternative methodology by projecting the knowledge base as an undirected weighted graph, referred to as the Mention-Entity Graph, consisting of nodes that are either candidate entities or entity annotations. Edges between entities on that graph model entity relatedness and edges between mentions and entities aim at encoding context similarities of the input knowledge base. The problem is then modeled as the search for a dense subgraph containing all mentioned nodes and only a single mention-entity edge for the disambiguated mentions; however, as this problem is NP-hard, an approximation algorithm was utilized. A similar technique was followed by [14], using an equivalent graph representation, mentioned as the Referent Graph, which presented corresponding primitives with the Mention-Entity Graph, but formed the disambiguation process core based on the PageRank algorithm.

Another approach in [15] introduces the HITS algorithm over a sub-graph of the RDF Knowledge Base, using a pruned breadth-first search traversal. The entities in the disambiguated text are used as initial seed nodes for the graph traversal. The system performance is achieved by the exploitation of coreference resolution heuristics, normalization, and expansion of surface forms.

In WAT [16], the successor of [12], the three-step architecture components were further refined and redesigned. Several methodology variations have been evaluated experimentally, including some basic graph-based methodologies, including PageRank and HITS. Individual experimental evaluations for each step of the disambiguation pipeline are included in the results, evaluating the performance of each individual component.

In [17], one of the first deep neural network models was introduced. The proposed methodology encodes representations of both mention, context, and entity in a continuous vector space. Then, a convolutional neural network models variable size context windows, embedding context word points on the vector space, and a neural tensor network is used for semantic modeling of context and mention interactions, in a compute-intensive process.

Ref. [18] also proposed an embedding method, mapping words and entities into the same continuous vector space. They extended the skip-gram model via a KB graph, and an anchor context model aimed to align vectors such that similar words and entities occurred close to one another in the vector space.

Another deep learning model was introduced in [19] employing neural representations. Entity embeddings, along with a “neural attention mechanism” over variable-sized context frames, are combined with a joint inference stage for the disambiguation step, in an approach attempting to combine some deep learning benefits with more traditional approaches such as probabilistic mention-entity maps and graphical models.

Apart from the most prominent research works outlined above, several similar endeavors in recent years approached Entity Linking and relevant AI tasks using deep neural network architectures, and faced similar challenges. However, vast dimensionality reduction due to the input dimension is required, introducing significant sense granularity deterioration at the input level as outlined in [20]. Furthermore, several recent works such as [21] question context-dependent deep neural network EL approaches in real-world big-data industry applications, as current methods focus increasingly on quality performance rather than run-time. To that end, RedW2 takes a step further back, introducing a context-free end-to-end Wikification system only utilizing knowledge of Wikipedia redirect pages, and outlining the feasibility of deep neural network approaches at scale.

## 2.2. Community Detection Algorithms

Due to the extensive use of hierarchical data and its profitable application in copious sectors in industry, community detection has attracted the interest of the scientific community. Therefore, by adopting diverse techniques and by following principles of various scientific backgrounds, a profuse amount of research has been conducted for tackling this NP-hard class problem. As Fortunato comprehensively presented [1], according to the evaluation strategy applied for unveiling the subjacent community hierarchy of any given information network, the existing methods can be distinguished into divisive algorithms, agglomerative algorithms, and other approaches.

By originally considering the given information network as a single community, the principle idea of divisive algorithms is to identify and remove the edges interconnecting different sub-communities by the maximization of a global network topology criterion at each repetition. Specifically, considering the input graph as a single community, a well-defined network topology metric, such as edge betweenness, vertex centrality, split betweenness, flow betweenness, edge centrality, and information centrality, is repetitively calculated for any pair of directly connected nodes. At each iteration, the edges that maximize, or minimize accordingly, the previously calculated topological measure are considered inter-connection edges and thus removed. This repetitive procedure finishes once no further edge removal is feasible. Due to the inherent stepwise revelation of finer communities, the execution evolution of these methods can be presented as dendrograms.

Among the ample number of divisive algorithms presented in [1], such as the Ratigan, Chen-Yuan, and Hole et al., the algorithm considered optimal in terms of execution performance and community extraction efficiency is that introduced by Girvan and Newman [1]. Particularly, in case of sparse

graphs, where the number of edges is linearly proportional to the number of nodes, this approach manages to identify the underlying community structure in  $O(mn^2)$  time complexity, where  $n$  and  $m$ , respectively, denote the number of nodes and the number of edges. The topological metric adopted is the edge-betweenness, which is practically the number of shortest paths that each edge engages per repetition, and mathematically demonstrates its information spread. Despite the undoubted robustness of the definition of edge-betweenness, the excessive requirements for repetitively calculating the shortest paths for all possible pairs of nodes inevitably limit the application of this algorithm to merely small information networks.

Contrary to the divisive algorithms, the agglomerative algorithms are bottom-up approaches that target an end result of a single community representing the input data graph. Specifically, by originally considering each node as an independent community, aka singletons, the ultimate goal of those methods is to repetitively merge the current step's communities by maximizing, or minimizing accordingly, a well-defined network topology-oriented similarity function. The formed meta-communities would apparently serve as the next iteration's communities. This iterative procedure ends if either a single community composed from the whole input data graph is formed or no further merging is possible. The extracted community structure's execution progress of the agglomerative algorithms can obviously be presented as a dendrogram, similarly to the divisive algorithm.

There are many noteworthy agglomerative algorithms [1], such as the InfoMap, the fast greedy algorithm, the label propagation method, the Kernighan–Lin algorithm, and the Xiang algorithm. However, the algorithm that is generally considered the state of the art in terms of apparent interpretation, plain implementation, eminent community extraction performance, and efficient execution, is the Louvain algorithm [2]. This method's key to success lies in the evaluation of any possible merge of directly connected communities using the modularity function that was formerly used as a quality assessment indicator of the final community structure. Specifically, the modularity similarity function deftly quantifies the concentration of the edges' intra-connecting nodes within a community, opposite to the corresponding nodes' degrees and the potential number of graph's edges, assuming a structure with no edges between the densely connected generated communities. Hence, by originally considering singletons, this greedy approach endeavors to compose meta-communities of communities by merging those for which the generated meta-community's modularity value is close to one, indicating a well-structured community, until no further merging is possible. Despite the fact that its exact computational complexity cannot accurately be calculated, this method seems to run in  $O(n \log n)$  time, as regards to the  $n$  number of included nodes.

A promising alternative to Louvain is the algorithm introduced by Clauset, Newman and Moore [3]. This heuristic method is also based on the iterative optimization of the modularity similarity function, although differs by:

- applying some predefined thresholds in the modularity gain on any possible merge,
- excluding all nodes of degree one from the original network and adding them back after the community computation and
- incorporating sophisticated data structures that simplify the hierarchical decomposition of the information network.

This method accomplishes the generation of community structures that are fundamentally different from Louvain's extracted hierarchies. Specifically, this algorithm tends to form meta-communities of a large fraction of nodes which have basically lower modularity values compared to the community structures identified by application of the Louvain algorithm. It is worth mentioning that in sparse graphs the computational complexity of this approach is assuredly limited to  $O(n \log^2 n)$  as regards the  $n$  number of nodes included.

Finally, inspired by radically different scientific backgrounds, the category of other approaches includes optimization methods that try to approximate this network topology optimization problem from substantially different perspectives. From methods inspired from discrete mathematics and



physics, to techniques that project the original community detection definition to conceptually different contexts, the conducted research can certainly be characterized affluent. The most indicative include, but are not limited to, the spectral optimization, the Wu and Huberman algorithm, the Information Diffusion Graph Clustering approach, Simulated Annealing, the Markov chains and random walks optimization algorithm, the genetic algorithms approximation, the External Optimization algorithm, and flow betweenness maximization based on the elementary Kirchhoff circuit theory. However, due to the over-demanding statistical analysis required and the mandatory data representation transformation applied, these are not further discussed and are considered outside the scope of this paper.

### 3. Materials and Methods

#### 3.1. Terminology and Notation

In this section the necessary terminology definitions are provided for readability convenience, as several notations have been established in previous works:

- Wikipedia pages are also mentioned as *Wikipedia entities*. A Wikipedia page is denoted  $p$ .
- Inter-wiki hyperlinks are cited as anchors, using the notation  $a$ . A text anchor linking a Wikipedia page from another page or text is referred to as *mention*.
- The first anchor of a text fragment is denoted  $a_0$ ,  $a_i$  the  $i+1$ th and so on.
- The anchor count of a text segment for annotation is denoted  $m$ .
- A Wikipedia page, as a mention (i.e., candidate sense) of the anchor  $a$  is denoted  $p_a$ .
- Due to natural language polysemy, an anchor may bare several mentions, hence the notation  $Pg(a)$  is used for the set of Wikipedia entities linked by anchor  $a$ .
- A set of Entities formulating a community as segmented by a community detection algorithm execution is noted  $C$
- The set of incoming links to Wikipedia Entity  $p$  is denoted  $in(p)$ .
- The cardinality of all Wikipedia Entities as in [9], is denoted  $|W|$
- Denoting the cardinality of the occurrences of an anchor text as mention in a text segment  $link(a)$ , and the total frequency of occurrences of that anchor, either as a mention or not,  $freq(a)$ , the link probability of a mention, aka commonness in related work, is cited as:

$$lp(a) = \frac{link(a)}{freq(a)}$$

The semantic annotation process carried out in the context of the introduced methodologies may be formally defined as the selection of a mention from  $Pg(a)$  for each identified anchor of the input text segment.

#### 3.2. Preprocessing

In this work, we leverage the rich set of semantic information that can be exploited from Wikipedia manual annotations by authors as our training set for our mention universe. In general, this universe of mentions expands along with the semantic coverage of Wikipedia. However, our method supports knowledge extension by incorporating Wikipedia annotated corpus in our training set. A limitation in our knowledge acquisition methodology would lie in the semantic coverage of Wikipedia entities. However, due to the crowdsourcing nature of Wikipedia, the creation of the entities and the editing of the knowledge of diverse and commonly accepted textual descriptions by a large number of editors has established Wikipedia Entities as a standard for the task, in a process known as Wikification.

As in [7,8,11,12,16] and several similar works, the set of mentions is extracted from an offline pre-processing of Wikipedia. Our methodology's universal set of entities consists of all Wikipedia articles in MediaWiki Main/Article namespace. A normalization needs to be carried out, so that all redirect pages are unified with their redirect target, including the necessary updates on Wikipedia

links to their target entity ids. Previously, this step involved following long redirect chains, however, this is remediated in recent Wikipedia snapshots.

As in [8,12,16], we utilized inter-wiki anchors, entity titles, and redirect pages for extracting a set of anchors that can be linked to a Wikipedia Entity, using the XML dump of Wikipedia as a sole source of input, after a single parse of the dump file. Maintaining the *anchor id*, *mention entity id*, and *source article id* as our input, and the corpus of Wikipedia dump, we efficiently can derive the tuple of available mentions (*anchor id*, *entity id*) including their occurrence prior-probability, also denoted as *commonness*, and the *anchor link probability*, denoted  $lp(a)$ . We applied filtering rules for pruning some low absolute and relative frequency mentions, along with some rules for discarding unsuitable for annotation anchors, such as stop-words and characters, from the mention universe (i.e., mention vocabulary). This is a common practice of similar works [9–16] that aims to maintain semantically significant mentions, while removing the unnecessary noise.

Specifically, the mentions dictionary contains roughly 37.5 million distinct anchors, containing 19.3 million entities which are interconnected in a link-graph of about 1.11 billion edges including edge enhancement via entity title redirect page processing.

As a next preprocessing step, we employ either Louvain or Clauset–Newman–Moore algorithm community detection on the Wikipedia Entity graph to achieve segmentation based on Wikipedia Entity community coherence characteristics. Hence, leveraging the ensured concept consistency, we manage to focus on a strict subset of Wikipedia Entities and thus improve the overall disambiguation efficiency by substantially demoting unmeaningful context from the collective mention consensus. Even if the computational complexity of these algorithms is  $O(n \log^2 n)$ , as outlined in [2,3] both algorithms' performance was evaluated in the context of the problem.

This offline task is required for the initialization process, as online updates can be applied on the fly leveraging a differential changes stream. Accordingly, for the community detection step, similar techniques may be applicable for on-the-fly community classification and rebalancing.

### 3.3. Mention Extraction

The first step of the text annotation task is anchor extraction. The set of mentions that can be recognized by our system (i.e., our mention vocabulary) is derived from the offline pre-processing of Wikipedia that leverages inter-wiki anchors, titles, and redirect pages for creating a database. Each such mention has a *link probability*, i.e., the probability of an occurrence of that mention as a link among all occurrences of that mention in Wikipedia. The link probability calculation is given by Formula (1). During the mention extraction step, the input text is tokenized composing n-grams of size up to six within the scope of our vocabulary as in similar other works [7,9,12,16]. Similar to the pruning applied to the mention universe, the n-grams attaining a link probability lower than 0.1% are filtered to discard stop words and unmeaningful mentions. In cases of overlapping n-grams, link probability is used as a strong indication of link-worthiness for the final anchor selection. Since the current work focuses on the value of community detection-based methodologies for the Wikification task, our evaluation is centered on the entity linking disambiguation step.

### 3.4. Disambiguation

Following the anchor extraction, the disambiguation process is involved in the resolution and selection of the best mention from  $Pg(a)$  for polysemous anchors identified in the input text segment. Unambiguous anchors may bare a single entity on their  $Pg(a)$  set, hence can be excluded from this process.

The intuition behind the novel methodologies of our contribution stems from the interpretation of Wikipedia using graph representation. We employ a novel scoring schema for the evaluation of candidate mentions of an anchor in a local context window. In general, it is anticipated that entities expressing a high degree of reciprocity would be also members of the same community. So far,

the exploration of relatedness metrics among Wikipedia entities has focused on the intersection of two articles' incoming links. As a result, transitive or deeper relations are neglected by the ranking process.

The evaluation score of each mention of an anchor during the disambiguation process is carried out by employing a Gaussian Process binary Classifier (GPC) [22] which was selected in terms of interpretability for assessing our methodology. Particularly, we model the entity linking task by performing binary classification on each probable mention of an anchor, for deriving probabilistic classification scores to the two obvious classes:

- the *entity linking compatibility* and
- the *entity linking incompatibility*

Thus, the predictions take the form of class probabilities. For our GPC implementation, the classifier approximates the non-Gaussian posterior with a Gaussian based on the Laplace approximation as detailed in Chapter 3 of [22]. Our classifier implementation used a Radial basis function kernel in all cases.

Following the terminology and notation presented in Section 3.1, the classification process is based on the following features:

$$lp(a) = \frac{link(a)}{freq(a)} \quad (1)$$

$$community\ coherence = 1 - \frac{\{max(|C|) : a_{0..k} \in C\}}{k + 1} \quad (2)$$

$$avg\ interwiki\ jaccard\ index(a_i) = \sum_{k=0}^{k=i-1} \frac{|in(p_{a_i}) \cap in(p_{a_k})|}{|in(p_{a_i}) \cup in(p_{a_k})|} / m + \sum_{k=i+1}^{k=m} \frac{|in(p_{a_i}) \cap in(p_{a_k})|}{|in(p_{a_i}) \cup in(p_{a_k})|} / m \quad (3)$$

$$avg\ relatedness(a_i) = \sum_{k \in \{p_{a_0} \dots p_{a_m}\} - \{p_{a_i}\}} \frac{\log(\max(|in(p_{a_i})|, |in(p_{a_k})|)) - \log(|in(p_{a_i}) \cap in(p_{a_k})|)}{\log(|W|) - \log(\min(|in(p_{a_i})|, |in(p_{a_k})|))} / m \quad (4)$$

Essentially:

- The *community coherence* presented by Formula (2) is used for modeling the coherence of mentions as members of common communities.
- The *average interwiki Jaccard index* of Formula (3) models the average intersection over the union of entity mentions in a text segment for annotation. This feature aims at introducing strong semantic coherence information. The Jaccard index is established by similar works such as WAT [16].
- The feature of *relatedness*, which generally models the Wikipedia entity semantic relevance, was introduced in [9] and broadly used in several works [9–12,16]. The generalized form of the *average relatedness* as proposed in (4) is the last feature for our binary classification model.

The binary output classes defined reflect the positive and negative linking compatibility within the current context based on the input features. The classifier is used for scoring each available mention of an anchor. The mention with the highest positive linking compatibility is selected at the final step.

Regarding the training stage, we may acquire knowledge from the Wikipedia corpus. However, training capability is not limited, and may include any generic Wikipedia Entity annotated corpora. In our case, we utilized the Wikipedia corpus for training our GPC binary model.

This methodology that combines a well-established set of features along with the newly introduced community coherence, enables transactive learning and provides valuable input during the training phase. Thus, counter examples can contribute on the appropriate identification of the correct sense of an ambiguous anchor within a context, allowing the exclusion of negative samples which would not superinduce considerable value.



### 3.5. Disambiguation Confidence

There are many cases where context provides insufficient underlying information for unambiguous disambiguation. Polysemy and underlying meanings may also be a characteristic of natural language phrases, sentences, etc.

Our disambiguation step computes a score for each mention of an anchor based on the surrounding context. At the output of the Gaussian Process Classifier, the probability estimates of the positive linking compatibility class can be leveraged for a disambiguation confidence score. Aiming at the quantification of uncertainty, we can successfully exploit a confidence score based on the relative difference of the next highest positive linking probability estimated mention. The overall methodology is also outlined in Figure 1.

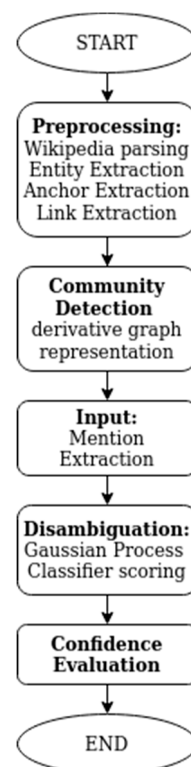


Figure 1. The proposed methodology flowchart.

## 4. Results

Our experimental evaluation was carried out using entities from Wikimedia Wikipedia dumps from 2019-12-20 [23]. Several raw database tables are available; however, we used the full copy of all Wikimedia pages enwiki-20191220-pages-articles [23], in the wikitext format of source and metadata embedded in XML as a single source of truth throughout our assessment. The efficient pre-processing of Wikipedia at its current size of 19.3 million entities interconnected in a link-graph of about 1.11 billion edges has been a challenging exercise. However, the employment of big-data techniques was critical for streamlining the ETL process. Due to the crowdsourcing nature of Wikipedia, there is a strong tendency for growth in the dataset over time, although our implementation's inherent distributed architecture and processing model would be capable of handling this big-data problem at a far larger scale.

For the experimental analysis, we focused on the established wiki-disamb30 dataset from [12] for evaluating the disambiguation process. This dataset consists of roughly 1.4 million short text segments randomly derived from Wikipedia pages and is generally considered as a baseline copious similar works [12,15,16,21]. Each text segment comprises approximately 30 words, including at least one ambiguous anchor and its respective Wikipedia article title annotation. As Wikipedia changes

over time, some preliminary pre-processing was required for either updating or filtering deprecated Wikipedia entities.

Our experiments involved a 64vCPU and 240 GB memory Google Cloud Platform setup, leveraging the scalability and parallelism of the Apache Spark framework during the ETL and pre-processing phase from the pages-articles.xml dump file. Robust implementations of Clauset–Newman–Moore [24] and Louvain [25] algorithms were leveraged for the two variants of our proposed methodologies. We adopted the `sclearn.GaussianProcessClassifier` [26] for our Gaussian Process Classification step.

At this point we should broadly emphasize the exceptionally vast context variability. Nevertheless, as the experimentation showed, the community detection (regardless of the algorithm applied) managed to efficiently partition the dataset into meaningful communities, with divergent community structures per case.

Regarding the disambiguation performance comparison, the classic prediction performance metrics are considered. Specifically, in reference to the entity linking concept, those can be outlined as shown below:

- Precision: The fraction of the correctly annotated anchors over the valid mentions.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

- Recall: The proportion of actual correctly annotated anchors.

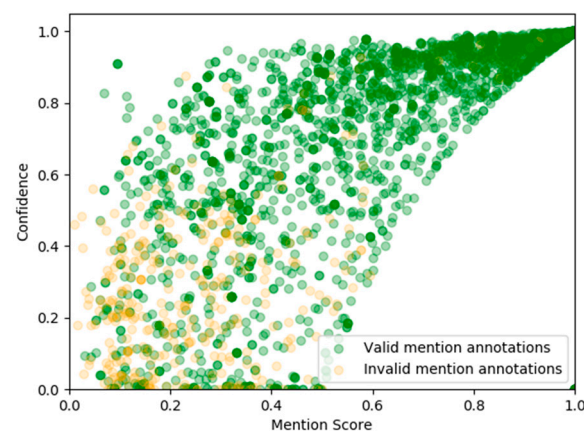
$$\text{Precision} = \frac{TP}{|\text{mentions}|} \quad (6)$$

- F1 Score: the harmonic mean of precision and recall.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

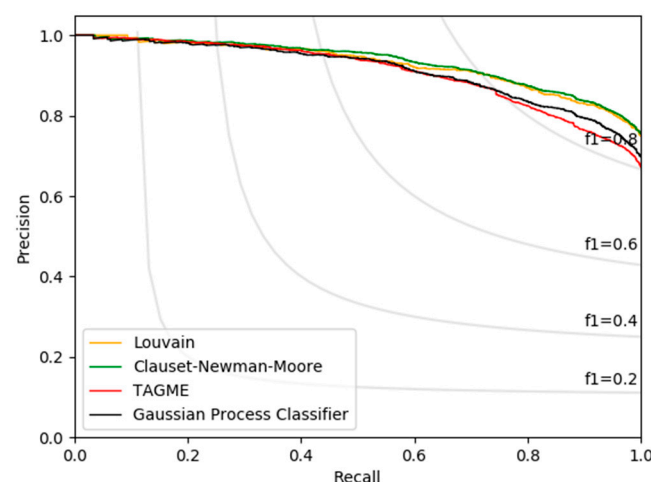
The assessment of the proposed methodology variants during an initial exploratory process contributed considerable insights. Similarly, the distinct community detection algorithms included in this work were more promising for further large-scale evaluation. We focused on the analysis of the entity linking disambiguation step, outlining the value of our *community-coherence* based methodology, and the review of the introduced confidence score as an indication of the disambiguation quality. Our experiments involved the evaluation of our described methodologies, in two variants. The first variant is based on the Louvain algorithm, and the second on the Clauset–Newman–Moore algorithm for the community detection step. As a comparative baseline, we included the performance of TAGME [12], along with the performance of our Gaussian Process Classifier methodology, without the consideration of the *community coherence* feature, to outline the contribution of the feature to the overall approach.

The Mention Score—Confidence scatterplot (Figure 2), outlines the correlation of the confidence score with the probability of a valid mention annotation as intuitively anticipated. As we model confidence, based on the relative difference of the next highest positive linking probability estimated mention, we can observe a proportionately insignificant number of invalid mention annotation outliers for some relatively high confidence scores. The generated GPC model, which considers *community coherence*, *link probability*, *average Jaccard index* of inter-wiki links, and *average relatedness* [9], yields impressive accuracy performance. Specifically, the proportion of the valid versus the invalid annotations appears highly inclined towards the valid annotations.



**Figure 2.** Mention Score—Confidence Scatter Plot.

As depicted in Figure 3 and Table 1, the precision-recall performance of both approximations is similar. Nonetheless, the Clauset–Newman–Moore algorithm practically proved slightly better when applied in the community detection step as also theoretically anticipated. Specifically, the Clauset–Newman–Moore algorithm is naturally expected to form few big communities and hence further retain the original ontologies’ network structure. On the contrary, the Louvain algorithm’s application predictably resulted in the generation of many smaller communities and hence further fragment of the initial network of ontologies. Despite the successful and consistent fragmentation of Louvain’s community structure, the nature of the problem fits best in the formation of larger discrete unified communities. This requires further analysis in future experimentation, since other community detection algorithms might bear even better performance from the examined baseline algorithms. In any case, and for any community detection algorithm applied, the adoption of the community coherence feature is practically proven to be superior for high precision performance. Specifically, in the extreme case of absolute recall, the precision difference of the methodology introduced reaches 5% per each baseline case as depicted in Figure 3 and Table 1. As a result, it is safe to conclude that even with the baseline application of a GPC classifier, the value of community prediction application is justified.



**Figure 3.** Precision-recall of the two community detection methodology variants, the Gaussian Process Classifier and TAGME [12] baseline.

**Table 1.** Precision, recall and F1 score of Louvain and CNM methodology variants.

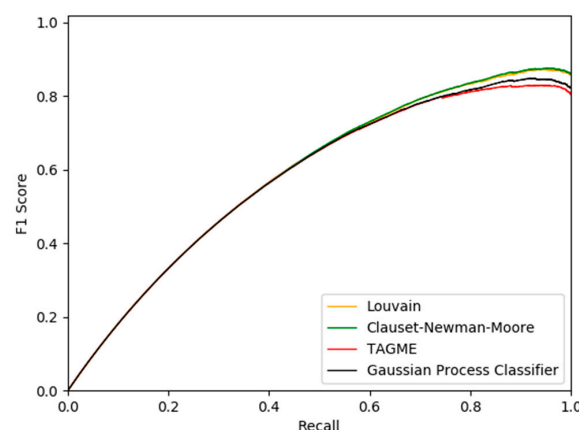
Recall	1.0	0.9	0.8	0.6	0.4	0.2
Louvain (Precision)	0.7491	0.8294	0.8693	0.9201	0.9643	0.9836
Louvain (F1)	0.8563	0.8632	0.8330	0.7262	0.5622	0.1809
CNM <sup>1</sup> (Precision)	0.7554	0.8362	0.8738	0.9337	0.9678	0.9866
CNM <sup>1</sup> (F1)	0.8606	0.8667	0.8351	0.7287	0.5659	0.3325
TAGME (Precision)	0.6720	0.7640	0.8242	0.9101	0.9619	0.9832
TAGME (F1)	0.8038	0.8264	0.8118	0.7222	0.5650	0.3323
GPC <sup>2</sup> (Precision)	0.6980	0.7946	0.8351	0.9108	0.9540	0.9808
GPC <sup>2</sup> (F1)	0.8221	0.8440	0.8171	0.7232	0.5633	0.3320

<sup>1</sup> Clauset–Newman–Moore, <sup>2</sup> Gaussian Process Regression without Community Coherence.

Furthermore, as the Gaussian Process Classification fits a generative probabilistic model, we observe a very gradual precision-recall curve slope. Overall, the contribution of community detection is remarkable as, even at very low confidence levels, i.e., when recall is over 0.9, the precision remains at exceptional levels, approximately 0.83. In Table 1, we see that even at a recall level of 0.8, both the Louvain and the Clauset–Newman–Moore algorithms attain impressive results for a dataset of this size and context variability, being highly propitious for real world applications. The performance divergence between the two examined algorithms is very low, outlining the fact that, despite community structure discrepancies, meaningful context information is contributing to the overall performance.

In general, the precision-recall curve of similar works [9–21] presents more aggressively steeper slopes in the (0.6, 1.0) range. Apparently, in the general case, the confidence level increase can be alternatively interpreted as requesting the absolute accuracy in the whole documents’ dataset. On the contrary, in our methodology the respective precision-recall curve is far smoother, proving the superiority of the introduced technique. This related to the fact that the mention retrieval occurs in the community level where the mentions belonging in the same community tend to have much more contextual similarity. Hence, as shown in the above graph, the higher precision-recall reported values are due to our methodology’s ensured community cohesion, which can be additionally translated to contextual resemblance.

In Figure 4, we observe the peak of F1 score near the 90% mark on the recall axis. Overall, the Clauset–Newman–Moore methodology attains consistently supreme performance, as clearly depicted towards the rightmost part of the recall axis. The nearly impeccable accuracy of our methodology, even at high recall, outlines the value of our approach to the problem, along with the substantially beneficial impact of introducing community coherence as structure interpretation of contextual resemblance in a highly variant collection of documents at a big-data scale.



**Figure 4.** F1 score of the two community detection methodology variants, the Gaussian Process Classifier and TAGME baseline.

## 5. Conclusions and Future Work

In this work we proposed a novel distributed methodology exploring the value of community detection algorithms for the text annotation problem, introducing a novel metric referred to as community coherence. An extensive experimental evaluation outlined impressive precision and computational performance. To the best of our knowledge, the current work is the first approximation of the Wikification problem employing Gaussian Process Classification and leveraging the value of community detection algorithms via the community coherence concept. As Bayesian approaches are established in text classification tasks, due to their inherent capabilities of probability distribution inference, our intuition in the successful application of the Gaussian Process Classification was validated by our experimental evaluation, yielding outstanding results. As the method is nonparametric, the training dataset observations are interpolated efficiently, as the problem is formalized effectively using a limited set of well-established features from previous works, mitigating the inherent drawbacks of the classification approximation, such as high dimensional space performance challenges.

Despite the promising results, there is a great room for improvement. Specifically:

- Increased interpretability was apparently required to assess the value and benefits of this novel methodology, however, the adoption of less interpretable but more accurate classifiers is to be considered a future step.
- The introduction of well-established features in a classifier-based entity linking methodology combined with community coherence resulted in impressive results, however, our low dimensionality approach could be leveraged by some deep neural network architecture models, as the next focus on further improvements to accuracy performance.
- The classic sequential community detection algorithms, such as the Louvain [2] and Clauset–Newman–Moore [3] algorithms, are doubtlessly considered unscalable and thus the experimentation with different community extraction techniques, such as the community prediction introduced in [27], needs also to be evaluated in terms of performance improvement and overall efficiency.
- The underlying challenge of the text annotation tasks, in conjunction with the Entity Linking tasks in general, would be the Knowledge Acquisition Bottleneck, namely the lack of semantically annotated corpora of the underlying knowledge ontology (i.e., Wikipedia). Addressing this challenge via an unsupervised machine learning approximation is among our future plans, aiming at improving knowledge acquisition closure from all available corpora resources.

Our introduced methodology is promising for large scale adoption. In this article, we primarily focused on the feasibility assessment of leveraging the value of Community Detection algorithms for the Text Annotation and Entity Linking task. Our exhaustive experimentation revealed auspicious results.

**Author Contributions:** Conceptualization, C.M., G.P. and M.A.S.; methodology, M.A.S. and G.P.; software, M.A.S.; validation, M.A.S. and G.P.; formal analysis, M.A.S. and G.P.; investigation, M.A.S.; resources, C.M.; data curation, M.A.S.; writing—original draft preparation, M.A.S. and G.P.; writing—review and editing, C.M., M.A.S. and G.P.; visualization, M.A.S.; supervision, M.A.S. and G.P.; project administration, C.M.; funding acquisition, C.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** Christos Makris has been co-financed by the European Union and the Greek national funds through the Regional Operational Program “Western Greece 2014–2020”, under the Call “Regional Research and Innovation Strategies for Smart Specialization—RIS3 in Information and Communication Technologies” (project: 5038701 entitled “Reviews Manager: Hotel Reviews Intelligent Impact Assessment Platform”).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174. [\[CrossRef\]](#)
2. Blondel, V.D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, 10008. [\[CrossRef\]](#)
3. Clauset, A.; Newman, M.E.J.; Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **2004**, *70*, 066111. [\[CrossRef\]](#) [\[PubMed\]](#)



4. Navigli, R. Word sense disambiguation: A survey. *ACM Comput. Surv.* **2009**, *41*, 1–69. [\[CrossRef\]](#)
5. Mallery, J.C. Thinking about Foreign Policy: Finding an Appropriate Role for Artificial Intelligence Computers. Ph.D. Thesis, MIT Political Science Department, Cambridge, MA, USA, 1988.
6. Gale, W.A.; Church, K.W.; Yarowsky, D. A method for disambiguating word senses in a large corpus. *Comput. Humanit.* **1992**, *26*, 415–439. [\[CrossRef\]](#)
7. Mihalcea, R.; Csomai, A. Wikify! Linking Documents to Encyclopedic Knowledge. In Proceedings of the CIKM 2007, Lisboa, Portugal, 6–8 November 2007; pp. 233–242. [\[CrossRef\]](#)
8. Silviu, C. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In Proceedings of the EMNLP-CoNLL 2007, Prague, Czech, 28–30 June 2007; pp. 708–716.
9. Milne, D.; Witten, I. Learning to link with wikipedia. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 509–518. [\[CrossRef\]](#)
10. Milne, D.N.; Witten, I. *An Effective, Low-cost Measure of Semantic Relatedness Obtained from Wikipedia Links*; AAAI Workshop on Wikipedia and Artificial Intelligence: Menlo Park, CA, USA, 2008.
11. Kulkarni, S.; Singh, A.; Ramakrishnan, G.; Chakrabarti, S. Collective annotation of Wikipedia entities in web text. In Proceedings of the 15th ACM SIGKDD International Conference, Paris, France, 28 June–1 July 2009; pp. 457–466. [\[CrossRef\]](#)
12. Ferragina, P.; Scaiella, U. TAGME: On-the-fly annotation of short text fragments (by wikipedia entities). In Proceedings of the 19th ACM International Conference on Information and Knowledge Management 2010, Toronto, ON, Canada, 26–30 October 2010; pp. 1625–1628. [\[CrossRef\]](#)
13. Hoffart, J.; Yosef, M.A.; Bordino, I.; Fürstenu, H.; Pinkal, M.; Spaniol, M.; Taneva, B.; Thater, S.; Weikum, G. Robust Disambiguation of Named Entities in Text. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 782–792.
14. Han, X.; Sun, L.; Zhao, J. Collective entity linking in web text: A graph-based method. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 25–29 July 2011; pp. 765–774. [\[CrossRef\]](#)
15. Usbeck, R.; Ngomo, A.-C.N.; Röder, M.; Gerber, D.; Coelho, S.A.; Auer, S.; Both, A. AGDISTIS—Agnostic Disambiguation of Named Entities Using Linked Open Data. In Proceedings of the ECAI 2014 21st European Conference on Artificial Intelligence, Prague, Czech, 18–22 August 2014; pp. 1113–1114. [\[CrossRef\]](#)
16. Piccinno, F.; Ferragina, P. From TagME to WAT: A new entity annotator. In Proceedings of the First International Workshop on Entity Recognition & Disambiguation, Gold Coast Queensland, Australia, 11 July 2014; pp. 55–62. [\[CrossRef\]](#)
17. Sun, Y.; Lin, L.; Tang, D.; Yang, N.; Ji, Z.; Wang, X. Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation. In Proceedings of the IJCAI 2015 Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 1333–1339.
18. Yamada, I.; Shindo, H.; Takeda, H.; Takefuji, Y. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. *arXiv* **2016**, arXiv:1601.01343.
19. Ganea, O.-E.; Hofmann, T. Deep joint entity disambiguation with local neural attention. *arXiv* **2017**, arXiv:1704.04920.
20. Sil, A.; Kundu, G.; Florian, R.; Hamza, W. Neural Cross-Lingual Entity Linking. *arXiv* **2018**, arXiv:1712.01813.
21. Shnayderman, I.; Ein-Dor, L.; Mass, Y.; Halfon, A.; Sznajder, B.; Spector, A.; Katz, Y.; Sheinwald, D.; Aharonov, R.; Slonim, N. Fast End-to-End Wikification. *arXiv* **2019**, arXiv:1908.06785.
22. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
23. Index of /enwiki/. Available online: <https://dumps.wikimedia.org/enwiki> (accessed on 5 May 2020).
24. Clauset-Newman-Moore Algorithm Implementation. Available online: [https://networkx.github.io/documentation/stable/\\_modules/networkx/algorithms/community/modularity\\_max.html#greedy\\_modularity\\_communities](https://networkx.github.io/documentation/stable/_modules/networkx/algorithms/community/modularity_max.html#greedy_modularity_communities) (accessed on 5 May 2020).
25. Louvain Algorithm Implementation. Available online: <https://github.com/Sotera/spark-distributed-louvain-modularity> (accessed on 5 May 2020).

26. sklearn.gaussian\_process.GaussianProcessClassifier. Available online: [https://scikit-learn.org/stable/modules/generated/sklearn.gaussian\\_process.GaussianProcessClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html) (accessed on 5 May 2020).
27. Makris, C.; Pispirigos, G.; Rizos, I.O. A Distributed Bagging Ensemble Methodology for Community Prediction in Social Networks. *Information* **2020**, *11*, 199. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).