Research Repository UCD

| | |
|---|---|
| ***Title*** | ANNOTATE: orgANizing uNstructured cOntenTs viA Topic labEls |
| ***Authors(s)*** | Ajwani, Deepak, Taneva, Bilyana, Dutta, Sourav, et al. |
| ***Publication date*** | 2018-12-13 |
| ***Publication information*** | Ajwani, Deepak, Bilyana Taneva, Sourav Dutta, and et al. "ANNOTATE: OrgANizing UNstructured COntenTs viA Topic LabEls." IEEE, December 13, 2018. https://doi.org/10.1109/BigData.2018.8622647. |
| ***Conference details*** | 2018 IEEE International Conference on Big Data, Seattle, United States of America, 10-13 December 2018 |
| ***Publisher*** | IEEE |
| ***Item record/more information*** | http://hdl.handle.net/10197/9888 |
| ***Publisher's statement*** | © 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. |
| ***Publisher's version (DOI)*** | 10.1109/BigData.2018.8622647 |

Downloaded 2026-05-01 23:36:32

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

# ANNOTATE: orgANizing uNstructured cOntenTs viA Topic labEls

Deepak Ajwani, Bilyana Taneva, Sourav Dutta, Patrick K. Nicholson, Ghasem Heyrani-Nobari and Alessandra Sala
*Nokia Bell Labs, Ireland*
{*deepak.ajwani,sourav.dutta,pat.nicholson,alessandra.sala*}*@nokia-bell-labs.com, biliana.taneva@gmail.com, qasem@sqnco.com*

***Abstract*—With the advent of Big Data paradigm, filtering, retrieval, and linking of unstructured multi-modal data has become a necessity. Assigning topic labels to contents, that accurately capture the meaning and contextual information, is a fundamental problem in organizing unstructured data. The usage of manually-assigned tags for this purpose introduces inconsistencies because of different "surface forms". On the other hand, existing automated approaches either use hierarchical multi-label classification, or are unsupervised and rely on (undirected) graph measures leveraging taxonomies. While the former requires large training data set to learn the characteristics of each topic class, the latter lacks the flexibility to learn broad range of related topics and are less accurate.**

**We propose a novel framework, ANNOTATE based on a small set of features and directed traversal of taxonomies to learn a broad spectrum of related topics using limited training data. We also show that our approach provides accurate labels for several domains without the need for re-training. For instance, the framework, trained on a small set of BBC news articles, exhibits close matches to user-generated tags for Quora documents. Experimental results, on the same model, for news classification and identifying aspects of Amazon product reviews, based on Amazon Mechanical Turk evaluation show our approach to be significantly better than state-of-the-art.**

**We further present real-life case studies of our proposed framework for automatically tagging Quora posts, and topically segmenting, indexing and linking related YouTube videos (using our publicly available Chrome browser extension).**

## I. Introduction

We live in an era of Big Data, where we struggle to keep up with ever-growing amount of information. It is estimated that a large part of this is unstructured content[1] – information that either does not have a pre-defined data model or is not pre-organized. This includes emails, chats, blogs, news and multimedia content. While unstructured data is available in abundance, organizing it for filtering and processing, by humans and machines, continues to be a major challenge.

A major difficulty in organizing unstructured content for future retrieval and insightful analytics is that it necessitates precise characterization of the topic of the content. Ideally, the tags or topic labels should facilitate the search and recommendation for other content on the same topic. More specifically, they should summarize the broader contents and overall theme of a document, catering to text summarization, topic-based search (e.g. [1]), keyword extraction (e.g. [2]), and document clustering (e.g., [3]). In essence, good topic labels should conceptually link the different content documents based on their topical coherence. Currently, a lot of the unstructured information is organized through manually assigned tags, where the tagging is done by the content-creator. Prominent examples include tags on question-answering platforms such as Quora and StackOverflow. However, the manual labels are not consistent as different people use different phrases to express the same topic and even when they use the same phrase, they may spell it differently. For instance, Figure 1 shows an example where a user gave "Life Lessons" and "Life and Living" as labels for a question, while other users gave similar questions the labels "Life Experiences" and "Learning." These inconsistencies result in similar content being assigned different labels, making it difficult to retrieve and recommend the relevant content. The problem is not restricted to subjective content, and even scientific contents suffer, where some users may label a post "speed", while others label it as "velocity".

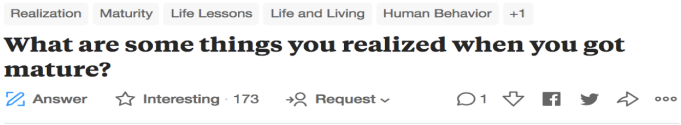



Figure 1: Question on social network Quora with inconsistent labels

To mitigate the effect of inconsistency in labels and organize the content better, some systems restrict the topic labels to be from a taxonomy and let the users agree upon good topic labels in a crowd-sourced way. Taxonomies provide a hierarchical representation of a rich set of topics and concepts, and enable easy navigation across general and specific categories. Large taxonomies are, thus, an ideal repository of canonical representation of good topic labels. An example of leveraging taxonomies in a crowd-sourced way is the assignment of category labels to Wikipedia pages. However, this is a time-consuming process as category assigners need to go through a range of possible categories, assign good labels to pages, argue over the different labels and continually update these labels. As such, this is useful only in a limited number of applications.

This problem becomes even more challenging for mul-

[1]https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/

timedia content, where the difficulties in organizing the content through tags are aggravated by the noise in speech-to-text techniques. Furthermore, multimedia content such as prime-time news, may cover different topics in different segments. Thus, we first need to segment the content before assigning labels to each segment separately.

As a result of these difficulties, a large amount of unstructured content remains unorganized and poorly analyzed. We have developed a cognitive platform, ANNOTATE (*orgANizing uNstructured cOntenTs viA Topic labEls*) that addresses these challenges head-on. ANNOTATE *automatically* organizes the unstructured content via a layer of canonical topic labels obtained from a taxonomy. These automatically annotated human-readable topic labels are neither too specific, nor too generic, thereby, enabling the retrieval of highly related content (without incurring topic drift). Furthermore, the features used in learning the topic labels are light-weight and ANNOTATE is able to compute the labels efficiently.

The system architecture of ANNOTATE consists of components to extract audio from a multimedia stream, convert speech to text, parse the text stream to identify noun phrases, disambiguate noun phrases to concepts in a taxonomy, segment the text stream if necessary, generalize the disambiguated concepts to identify a good topic label and conceptually linking the document segments to facilitate the navigation of multimedia documents. ANNOTATE has a modular design and it allows different techniques to be used in different components.

One of the most innovative component in ANNOTATE is the generalization of disambiguated concepts to a higher level category that acts as a topic label. ANNOTATE leverages the Wikipedia category taxonomy for this purpose, as it is broad and covers many millions of concepts and category labels. Furthermore, this taxonomy is regularly updated as new concepts and events emerge and this extension can even be done automatically with fairly high accuracy [4]. Note that ANNOTATE is not restricted to this taxonomy – The computed features are agnostic to the taxonomy used and thus, other taxonomies such as YAGO [5] or domain-specific taxonomies can be easily substituted. From the very large (order of millions), but fixed set of category labels in the taxonomy, ANNOTATE is able to find a small set of labels (5-10) for each document that precisely capture its content.

Existing approaches for the problem of topic labeling can be divided into two categories: (i) supervised techniques for hierarchical multi-label classification [6], [7], [8], [9], and (ii) unsupervised methods using undirected graph metrics leveraging taxonomy structures (e.g., [10]).

Supervised approaches use training on wide variety of features like word frequency, n-grams, language models, co-occurring mentions and text matches for hierarchical multi-label classification to associate documents labels [11], [9], [12]. Although such techniques have been shown to perform well in practice, they require enormous training data, and suffer from robustness and re-training issues. In contrast, ANNOTATE is robust in extracting a wide range of related topic labels, requires limited training data and can seamlessly cater to diverse applications without re-training, enabling "global semantic linking" across domains.

Unsupervised techniques view the taxonomy as an *undirected* graph and use undirected graph measures (e.g., centrality [10], clustering [13], PageRank [14]) to rank topics. Such approaches tend to lose the subsumption hierarchy and induce topic drift, generating labels of low descriptivity, providing a poor indicator of semantic relatedness. In contrast, ANNOTATE views the taxonomy as a directed acyclic graph (DAG) and utilizes *directional traversal* measures for information propagation and topic containment, improving label accuracy by using efficient features as a *proxy to capture semantic relations*.

**Contributions.** Our main contributions are:
(1) A key breakthrough in ANNOTATE that enables the computation of highly accurate topic labels efficiently is the identification of a small set of discriminating features that are adept in modeling graph-theoretic, information-theoretic, content-based, and other existing measures.

(2) Empirical results on three large real-life datasets (Amazon reviews, Quora posts, and BBC news articles) show that ANNOTATE achieves better accuracy. Evaluations using Amazon Mechanical Turk show that the topic labels generated by ANNOTATE are significantly more descriptive and better capture the related topics (sifting out unrelated or overly generic topics) than state-of-the-art approaches.

(3) We further present case-studies on tagging Quora posts and indexing YouTube videos to show the potential of ANNOTATE in automatically organizing vast swathes of unstructured content. Note that our YouTube indexing system is publicly available as an extension of the popular Chrome browser [2]. Also, we plan to make our datasets and labels public in the hope that this large dataset, annotated with correct and incorrect topic labels in a crowd-supervised way, will become an important benchmark in the community, accelerating further research in the area.

## II. Related Work

**Extracting Labels from Text.** Traditional approaches involved the extraction of the most likely label from the text itself (e.g., [15], [16], [17]). However, such approaches miss out on good labels (particularly for short text) as the underlying assumption that the best label is present in the document is, often, not true. Although, [18] proposed to extend the terms extracted from text by querying Wikipedia (using topic terms) and adding n-grams from the returned articles, the set of obtained topic labels remains restrictive.
**Topic Modeling.** Topic modeling techniques such as Latent Dirichlet Allocation [19], probabilistic Latent Semantic

[2]chrome.google.com/webstore/detail/npkafmpocoljkekkbccdkacmjpacoljd

Analysis [20], and the work of [21] model document contents as a mixture of topics, where each topic is a probability distribution over a bag-of-words. These approaches do not disambiguate entities for candidate labels, but primarily leverage raw text. We extract labels directly from taxonomy.

**Hierarchical Multi-label Classification.** The problem of associating multiple labels is referred to as *multi-label classification* [22], [23]. A basic approach to multi-label classification involves training a classifier for each of the candidate labels independently [24]. However, when the number of labels is very large (e.g., in thousands) such an approach becomes infeasible. Multiple approaches have been proposed to alleviate this problem by applying label transformations and dimensionality reduction [25], or randomized sampling of a set of labels [23]. Although the former enables labeling from a larger set of potential labels, they are infeasible for labeling contents with arbitrarily large topic set. The latter restricts the topic labels to a sampled set resulting in poorer topic labels (with larger topic drift) with loss of important connotations. Note, the set of potential topic labels are not totally independent, but are semantically related within a hierarchical structure. ANNOTATE leverages concepts from large hierarchies (Wikipedia concept taxonomy has more than 5 million nodes) for content labeling.

A related problem is hierarchical multi-label classification, wherein documents are categorized with labels organized in the form of a tree or a directed acyclic graph (DAG). Preserving the hierarchy was re-formulated as optimal subgraph finding in the tree or DAG by limiting the number of labels with dimensionality reduction [7]. Several approaches based on learning a binary decision tree for each label and learning a single decision tree for all labels was presented in [6]. Structural SVMs for classification of short documents in social streams was proposed in [9], while a kernel-based algorithm and dynamic programming was studied in [8]. However, these approaches either reduce the number of topic labels resulting in poorer labels or require large training data to learn discriminating features for each label. We note that unlike the case of disambiguation (that we use as a component), publicly available large datasets for topic labeling are scarce.

**Graph-based Approaches.** Unsupervised methods using undirected graph centrality measures to rank the label nodes was proposed by [10], adapting traditional centrality measures such as betweenness centrality [26], closeness centrality [27], information centrality [28] and random walk betweenness [29] to focus on paths between the set of entity leaves. A graph-based clustering approach by connecting similar comments to form an undirected graph was developed in [13] to extract central entities as topics. Furthermore, keywords extracted by topic models for querying on search engine to create an undirected graph from words contained in search results were also explored. Undirected PageRank algorithm was used to assign weights to words and score candidate labels [14].

**Information Content.** An information content measure for semantic similarity in taxonomies was proposed by [30], providing an estimation of the degree of generality versus the concreteness for a given label. A modified version of the information content measure was proposed in [31]. Further, text classification approaches [32] expand entities/concepts present in documents by adding synonyms, 1-hop ancestor, etc., to provide an enhanced text similarity measure. However, observe that topic labeling is different from binary classification or clustering of texts, and is related to multi-label categorization from a very large number of potential labels. Existing methods suffer from the need of huge data for training on all label instances and different content types.

## III. System Architecture

The ANNOTATE system consists of a highly modular and robust pipeline. Various techniques can be used as plug-and-play in different components of this pipeline without affecting the other components. The pipeline consists of the following components:

- Extract audio from a multimedia stream and convert speech to text: In recent years, many speech-to-text systems have been developed that achieve a very high accuracy[3] and can be used for this part.
- Parse the text stream to identify noun phrases: To identify noun phrases correctly, ANNOTATE utilizes a carefully defined set of rules. In addition, it can also leverage annotations provided by disambiguation systems (such as TagME [33]).
- Disambiguate noun phrases to concepts in a taxonomy: In recent years, there has been considerable work on this topic and many good disambiguation systems have been developed. We note that the disambiguation system from [34] is tightly integrated into ANNOTATE for supporting fast queries. However, ANNOTATE can leverage other disambiguation systems as well (e.g., TagMe [33]).
- Segment the multimedia stream: To segment a multimedia content, it is first divided into a large number of small duration (30 sec - 1 min) clips. These clips are then merged based on the semantic similarity of the corresponding transcripts. The semantic similarity can be computed either based on the word embeddings of noun phrases in the transcripts of the two clips or based on the intersection of in and out-links in Wikipedia pages corresponding to disambiguated concepts extracted from the two clips. An agglomerative merging procedure provides topic-coherent segmentation of multimedia content.
- Prune the set of disambiguated concepts: One big difficulty in labeling is that only a small fraction of

[3]For a recent survey of good software in the area, we refer the reader to https://windowsreport.com/speech-text-software/.

the entities disambiguated from the text are actually covered by the main topic. Even though the topic labeling component of ANNOTATE is robust, one still needs to prune the set of concepts that are likely to be covered by the main topic and only pass these concepts as input to the topic labeling component. To do this, ANNOTATE clusters the disambiguated concepts based on their semantic similarity and only uses the concepts in the largest cluster for topic labeling. Once again, the semantic similarity can be estimated through a common embedding of noun phrases and disambiguated concepts or it can be estimated based on the intersection of in-links and/or out-links of the Wikipedia pages corresponding to the disambiguated concepts.

- Generalize the disambiguated concepts to identify a good topic: The disambiguated concepts are then generalized using a machine learning solution based on a learning-to-rank paradigm. This solution leverages carefully defined features to find the appropriate level of granularity for the topic categories in a taxonomy. This component is described in detail in Sections V and VI. A detailed empirical study for this component has been presented in Section VII.
- Conceptually linking the document segments to facilitate the navigation of multimedia documents

## IV. Preliminaries

**Taxonomy.** A taxonomy is a directed graph $T = (V, E)$ where the node set is the union of the entities and concepts from a knowledge base. Edges represent relationships between the entities and categories, as well as between categories. The direction of an edge $(u, v)$ indicates that $v$ is a broader category of $u$, and entities (which cannot be further refined) therefore have in-degree zero. Though $T$ is not necessarily a tree, we refer to entities as *leaves*.

As the backbone taxonomy, ANNOTATE uses the Simple Knowledge Organization System (SKOS) formatted categories, over Wikipedia articles, available through DBpedia [35]; wherein the relationships between the articles and categories are captured by the "skos:broader" relationship. In this graph, each Wikipedia entity is a leaf, and can have multiple outgoing edges to categories. Categories also have out-edges to other categories, as shown by the example subtaxonomy in Figure 2.

We refer to the set of nodes reachable in $T$ from an arbitrary node $u$ as the set of ancestors of $u$, denoted as $\mathcal{A}(u)$. If $S$ is a set of nodes, then $\mathcal{A}(S) = \cup_{u \in S} \mathcal{A}(u)$. Similarly, for a category node $u$, we refer to the set of leaves $\ell$ such that $u \in \mathcal{A}(\ell)$ as the descendants of $u$ (denoted as $\mathcal{D}(u)$) and extend the definition to sets of nodes analogously. Finally, we consider neighbourhoods around sets of leaves. For a node set, $S$ in $T = (V, E)$, let $N^1(S)$ denote the *1-hop neighbourhood* of $S$ (i.e., $\{v : \exists u \in S, v \in V \ ((u, v) \in E)\} \cup S$). Similarly, define $N^k(S)$ , the *k-hop neighbourhood* of $S$ as $N^1(N^{k-1}(s)) \cup N^{k-1}(S)$ for all $k \geq 2$. In Fig. 2, if $S = \{\text{Border Gateway Protocol}\}$, then $N^1(S) = \{\text{Routing protocols}, \text{Internet protocols}\} \cup S$, and $N^2(S) = \{\text{Network protocols}, \text{Routing}\} \cup N^1(S) \cup S$.

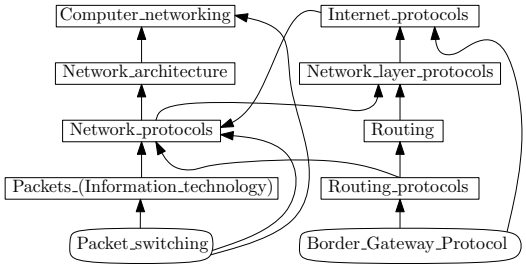



Figure 2: Taxonomy Subgraph. (Nodes inside rounded rectangles are entities, while nodes inside rectangles are categories.)

**Removing Cycles.** The graph $T$ may contain cycles [36], [37], [38], [39] due to errors from crowd-sourced data collection or from automated merging of data sources[4]. For example, Fig. 2 contains a cycle of length three with categories: *Network layer protocols, Internet protocols, and Network protocols*. Since cycles are inconsistent with the logical usage of taxonomies (modeling broader-specific relation among nodes), we clean the taxonomy by removing cycles as a pre-processing step (e.g., using a DFS-based technique in line with [37] or using a learning model [40]), to obtain the final taxonomy graph $T$, a directed acyclic graph (DAG).

## V. Measures of Label Quality

As input, we are given a text document and a set $S$ of extracted entities with associated weights $w(e)$, where $w(e)$ represents the frequency of $e$ in the input text. In addition, given a taxonomy DAG $T$ (Wikipedia in our case), our goal is to extract labels from $T$ that characterize the set of entities $S$ and accurately reflects the contents and context of the input document.

ANNOTATE capture a range of properties incorporating graph-theoretic, set-theoretic and text-similarity based features to define "good topics". Specifically, ANNOTATE navigates $T$ to compute a novel set of features capturing: (a) *centrality measures* based on novel notions of random walk leveraging directional traversal of taxonomy, (b) *information propagation* to obtain weights for entity-topic relationships, (c) *text-similarity* on word distributions between topics and documents, and (d) *precision* and *coverage statistics* between topics and entities. These features were carefully selected from a set of more than thirty measures by considering them as features for a random forest classification model. Next, we discuss the proposed measures, and, later, we show that ANNOTATE extracts quality topic labels.

[4]SKOS does not require the graph to be cycle free, which can be a problem in some applications: www.w3.org/TR/skos-reference/#L2484.

**(1) Information Propagation:** Intuitively, a candidate topic node is precise if it generalizes most of the entity leaves present in the document, but not other entities. We model this intuition using an information propagation (IP) measure, wherein weights of entity leaves are propagated to topic nodes in the DAG, $T(V, E)$. To prevent over-generalization, each intermediate node decays the propagation of weights by a multiplicative factor of $(1-\delta)$ , i.e., only $(1-\delta)$ fraction of weight passes through a node. Thus, an intermediate node exhibits high propagation if there exist many short directed paths to it, corresponding to its generalizability.

Let $l(v)$ be the number of parents of a node $v$ and $f(v)$ be the frequency of the corresponding entity in the document. Initially, the weight of an entity leaf $v$ is equal to its frequency $f(v)$ in the document and the weight of a non-leaf node is zero. The output score, $\text{inf}(v)$ of a topic node $v$ is the total weight propagating through it. Thus,

$$\text{inf}(v) = \begin{cases} f(v) & v \in S \\ \sum_{(u,v)\in E} \frac{\text{inf}(u)*(1-\delta)}{l(u)} & v \notin S \end{cases}$$

This measure can equivalently be seen as a random walk traversal (starting from an entity towards the root) over the taxonomy. Unlike traditional notions of random walk, where steady-state probabilities are used to rank nodes, we rank the nodes using the expected number of random walks terminating there. Note that, if $f(u)$ random walks start from $u$, the expected number of random walks that terminate at node $v$ is equal to $\delta \times \text{inf}(v)$. Since $\delta$ is a constant, ranking based on information propagation score or random walk termination probability is equivalent. Also, since the taxonomy is a DAG, we do not need a iterative procedure to approximate the probabilities for the random walk, and the measure can be computed *exactly* in one pass.

However, noisy crowd-sourced taxonomies may deteriorate the IP quality due to high variability in the number of parent categories assigned to a node. As, for nodes with many parents, the propagated weights become thinly divided among parents, while for nodes with only one parent $(1-\delta)$-fraction of weight is passed along. Further, such taxonomies might contain edges from very specific to highly general nodes, resulting in generic nodes getting a higher score than expected.

To ensure robustness against such taxonomy imperfections, we modify the above measure as:

1) Rather than dividing the undecayed score among the $l(v)$ parents, we pass a larger fraction $1/l(u)^{\alpha}$ of the weight to each parent for some $\alpha < 1$.
2) Generic nodes having longer paths to leaf are penalized by a penalty factor governed by constant $\beta$.

Let $TO(v)$ be the longest path length to $v$ from a leaf node in $T$. The modified IP score of node $v$ is then,

$$\text{inf}(v) = \begin{cases} f(v) & v \in S \\ \sum_{(u,v)\in E} \frac{\text{inf}(u)*(1-\delta)}{l(u)^{\alpha} * e^{TO(v)*\beta}} & v \notin S \end{cases}$$

Picking a higher value of $\alpha, \beta$ or $\delta$ would favour topic nodes closer to the leaves and penalize categories that are very generic. Based on empirical evidence, we set $\alpha = 0.9, \beta = 0.005, \delta = 0.05$ for our experimental evaluations on the Wikipedia-SKOS taxonomy. The calculation of information propagation scores is done in topological order, from leaves to general topic nodes, similar to the random walk computation on $T$ and $T'$.

**(2) Closeness Centrality to Entity Leaves:** A good topic label should have a small average distance to the entities of the document. Since a large fraction of entity leaves is unreachable, we use the closeness centrality, defined as the inverse of the average *undirected* distance to the leaves. Note that this is similar to the focused closeness centrality measure of [10], except that their measure considers 2-hop neighbors of $S$. ANNOTATE considers a larger subgraph of the Wikipedia taxonomy, making the resultant graph almost always connected and providing a better approximation of its centrality.

**(3) Overlap and (4) Consistency:** A topic label, $l$, is considered a "good" candidate for content, $D$, if there exists a high degree of overlap between the entities present as descendant of $l$ and the set $S$ for $D$. This reflects a high potential similarity between the context of the document and that of the label. In order to capture this interaction, we define the *Overlap* measure as the cardinality of intersection between the descendant leaves of a topic and the entities mentioned within the text document. Formally, we define $\text{Overlap} = |\mathcal{D}(\ell) \cap S|$.

Topic labels that are too specific (limiting content retrieval and linking) do exhibit a low *Overlap* value. However, a very generic label, also considered to be bad, would demonstrate high overlap, and be considered as a viable candidate. To address such cases, we propose the *Consistency* measure to capture the generality of a topic.

An appropriate topic should not only exhibit high overlap value (i.e., high recall or identification of document entities), but also provide a low level of reachability (in descendant leaves) to unrelated concepts. This reachability constraint is captured by a measure of precision between the set of descendant concepts of a topic (from the taxonomy) and the content concepts. Similar to the F1 score, we define *Consistency* as the harmonic mean of precision and recall for a candidate topic label,

$$\text{Consistency} = \frac{2PR}{P+R}, \text{ with } P = \frac{Overlap}{|\mathcal{D}(\ell)|} \text{ \& } R = \frac{\text{Overlap}}{|S|}$$

**(5) Document and (6) Entity Similarities:** A topic label is considered as good if the document and the topic description have highly similar contents, or the description of entities and topics are similar. The associated Wikipedia pages of concepts/topics are considered as their description (filtering out labels with no Wikipedia pages) for defining two text-similarity measures: (a) *tf-idf cosine similarity*

between the document and the Wikipedia topic page abstract, and (b) tf-idf similarity between Wikipedia abstracts of entities and topics.

## VI. Classification Model

The above $six$ measures are used as features for training a *Random Forest* based classifier [5] in ANNOTATE to learn "good" and diverse topics reflecting the information spectrum of the document as well as reducing topic drift. Akin to the point-wise learning-to-rank paradigm [41], [42], we learn a mapping from the feature vectors to binary labels of a topic (whether good or bad), which is then used to predict the labels for the test documents. Since the feature set is small (total of 6 features) and we use a restricted candidate topic label search, our model efficiently leverages taxonomies with millions of topic labels using few training examples that can be easily obtained using crowd-sourcing. Further, our approach requires significantly less training data since we are only interested in learning the parameters to combine the different measures and not the individual characteristic of (millions of) topic labels. Specifically, we extract a set of *candidate labels* for entities in $S$, and consider the nodes in $\mathcal{A}(S)$ – the set of ancestors of entities in $S$ in $T$ – as potential candidates for labeling $S$. To select the best label, a ranking score for each label $\ell \in \mathcal{A}(S)$ is computed reflecting the different measures, and is combined using a supervised approach. Specifically, we use an implementation of *random forest classifier* from the python scikit-learn library (scikit-learn.org/stable/).

## VII. Experimental Evaluation

**Training Setup.** The *training corpus* for ANNOTATE (for its binary classifier) consisted of $50$ selected BBC news articles [6] from [43] each truncated to around 150 words. We used the TagMe [33] web API[7] to extract entities from the training corpus and disambiguate them (with TagMe parameter $\rho = 0.1$) to Wikipedia articles, forming the set of leaves, $S$ for our taxonomy $T$. Although the performance is dependant on the accuracy of entity disambiguation; popular tools have shown to be highly accurate (78% by [33]). Further, since the number of entities in medium-sized documents is large, a few errors in disambiguation do not severely affect our feature set as shown later.

For each text in our training corpus, we generated candidate labels by sampling $10$ nodes uniformly at random without replacement from the following sets: $S$; $N^1(S)$; $N^2(S) \setminus N^1(S)$; $N^3(S) \setminus N^2(S)$; and $N^4(S) \setminus N^3(S)$. Overall, in this pre-processing step we extracted $2474$ labels for training[8] the learning-to-rank framework.

We used topic labels from Amazon Mechanical Turk (AMT) [9] to establish a ground truth data set using the extracted candidate labels. For each document in the training corpus, we created a set of Human Intelligence Tasks (HITs), where a set of workers were provided with texts and up to $10$ candidate topic labels, and were asked to remove the unrelated topics. In total, $250$ HITS (5 categories, 10 texts per category, each with 5 hits) were created exhaustively, with each extracted label appearing in exactly one HIT, and each HIT was assigned to five workers. Using the responses, we divided the labels as:

1) *Good*: labels that $0$ or $1$ worker marked as incorrect.
2) *Ambiguous*: labels marked wrong by $2$ or $3$ workers.
3) *Bad*: labels marked incorrect by $4$ or $5$ workers.

Ambiguous labels were discarded, and a value of $1$ was assigned to good labels, and $0$ to bad labels. We obtained $1898$ ground truth labels with $572$ good and $1326$ bad labels (annotated data would be publicly released upon acceptance). AMT workers with $> 98\%$ HIT Approval Rating and $> 5000$ approvals were used for trustworthiness, and accuracy of workers was checked by qualification test on $10$ ground truth questions (few responses were manually checked). Further, worker bias was reduced by majority-voted good/bad topics usage.

**Classification Model.** After training, we performed 10-fold cross validation of ANNOTATE (with documents from 5 categories randomly shuffled) using the collected ground truth data. We experimented with various classification models (SVMs with different kernels, Ada-Boost Trees, etc.), to find *random forest* to exhibit the best cross-validation accuracy. Random Forest reported an ROC of $94\%$ compared to $86\%$ achieved by SVM. We also found that the accuracy improved upon increasing number of estimator trees in the random forest, reaching a plateau around $250$ trees. Hence, we use $250$ trees for our experimental setup, and set the minimum number of samples required to split an internal node to $5$, as lower values resulted in over-fitting and poor accuracy. The mean scores and 95% confidence intervals obtained were: *ROC AUC - 0.94 ($\pm$ 0.04); F1 - 0.81 ($\pm$ 0.14); Precision - 0.86 ($\pm$ 0.13); Recall - 0.78 ($\pm$ 0.15); Accuracy - 0.89 ($\pm$ 0.07)*. The results showcase the accuracy of the classification model in ANNOTATE , with a large area under ROC curve, an F1 score of $0.81$, and accuracy of $0.89$. The feature importances were as: (5) Doc-Topic similarity (0.280), (1) Information Propagation (0.270), (6) Avg. Entity-Topic similarity (0.169), (2) Closeness Centrality (0.109), (3) Overlap (0.088), and (4) Consistency (0.084).

**Candidate Parameter.** The computation time to extract topics increases proportionally with the number of candidates, which in our case are all ancestors (in Wikipedia taxonomy) of entities in the document. Experimentally (omitted due to space constraints) we observed that restricting candidate

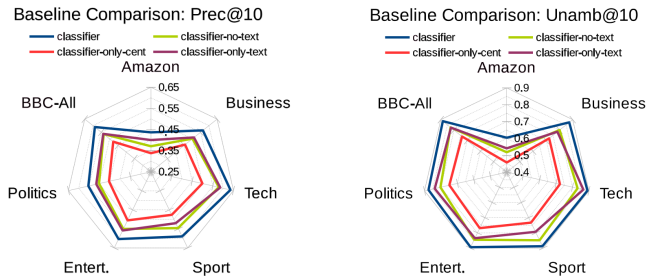
[5] We also experimented with other classification models such as SVMs with different kernels, Ada-Boost Trees, etc.

[6] From mlg.ucd.ie/datasets/bbc.html with 10 articles from 5 categories: Business, Entertainment, Politics, Sports, & Technology.

[7] https://tagme.d4science.org/tagme/

[8] Sometimes number of candidate labels obtained is less than 10.

[9] (www.mturk.com/mturk/welcome)




(a) (b)

Figure 3: Spider-plot of results attained by the different baselines for (a) Prec@10 and (b) Unamb@10 on the different data sets.

labels to set $N^3(S)$ provided a good trade-off between accuracy and run-time. Specifically, only 10% of the ground truth labels (in Quora) were missed by considering $N^3(S)$ (with comparable Prec@10 to $\mathcal{A}(S)$), while reducing computation by an order of magnitude. We use this setting in our evaluation.

**Test Datasets.** We evaluate the performance of the classification module of ANNOTATE on 3 real datasets:

1) 500 news articles from the BBC corpus [43], *disjoint* from the training corpus[10].
2) 500 electronic product reviews from Amazon,[11] for a variety of audio and video equipment.
3) 1067 Q&A posts from Quora on 14 topics[12].

Note that because of the costs involved with using AMT workers for validating annotations, a certain self-restraint on the size of the test data was unavoidable.

As in the training step, the documents were truncated to around 150 words, and TagMe [33] was used to extract and disambiguate phrases. Our experiments are divided into 2 categories – *human annotated* (for Amazon & BBC), and *direct comparison* (for Quora).

For the human annotated experiments, candidate labels for each test document was ranked based on the score from our classification model and competing approaches – and for each approach, the top-10 labels were extracted. Similar to Sec. VII, a set of HITs for each text were created on AMT, each HIT presented to five workers, and the annotated labels obtained grouped into *good, ambiguous, and bad* categories.

For direct comparison experiment, we focus on the user generated labels on Quora that exactly match Wikipedia page titles. We found 73.61% of Quora labels to be present in Wikipedia with an average of three labels per document, and treat them as the ground truth labels.

**Comparison.** We compare our proposed ANNOTATE (i.e., the classification module, *classifier*) (Sec. VI) against the following approaches and baselines:

**(1)** *classifier-only-cent* [10]: A random walk betweenness centrality measure, focused on entity leaves, was shown to achieve the best topic label accuracy on various datasets. A classification model was learnt on the training data (of Sec. VI) using the centrality measure.

**(2)** *classifier-only-text* [44]: Text-similarity based classification model using text similarity features of Sec. V.

**(3)** To gain additional insights, we perform ablation study with: *classifier-no-text* – classification model without text similarity, and *classifier-no-text-no-cent* – classifier without text similarity and closeness centrality.

**Evaluation.** We use the following performance metrics:

*Prec@10:* The micro-averaged *precision* measures the ratio of the number of good labels to the number of labels generated. Each approach (generating 10 labels per document) showed similar macro-average results.

*Unamb@10:* The *unambiguous precision* is the (micro-averaged) ratio of the number of "clear" labels to the number of *unambiguous* labels (i.e., sum of good and bad labels). This paints a clearer picture on label quality – larger difference between precision and unambiguous precision shows higher disagreement among the workers.

*Accuracy:* It is the ratio of the number of times ($g$) that all the labels generated were unanimously marked as good by the annotators, to the product of the number of annotators ($k$) and the total number of labels ($t$); i.e., $g/(kt)$. This further quantifies the quality of a topic based on the agreement among annotators taking into account both, the label types and its score.

*NDCG:* For Quora dataset, we report the *normalized discounted gain* (NDCG) computed between the ground truth label list and labels output by the approaches, ranked by score. Since the labels are not ordered, they were considered equally important ($relevance = 1$). Thus, for document $\mathcal{D}$ with $k$ ground truth labels, $\ell_1, \ldots, \ell_k$, the discount cumulative gain (DCG) is: $\text{DCG}(\mathcal{D}) = \sum_{i=1}^{k} \frac{1}{\log_2(p(\ell_i)+1)}$, where $p(\ell_i)$ is the rank of label $\ell_i$ in the output label list. $NDCG$ is $DCG$ divided by *ideal discounted gain*, which is $\sum_{i=1}^{k} 1/(\log_2(i+1))$. We report both micro- and macro-averaged NDCG.

**Results.** Table I tabulates the results obtained by human annotated AMT experiment on Amazon and BBC. We observe that our proposed framework, ANNOTATE outperforms the competing approaches on both datasets, with *statistically significant*[13] performance improvement of around 6% on all metrics.

Figure 3 presents spider chart results on Prec@10 and Unamb@10 measures for Amazon and BBC. We observe ANNOTATE to outperform the other approaches on the entire

[10]We selected 100 articles from each of the 5 categories: Business, Entertainment, Politics, Sports, & Technology.

[11]https://snap.stanford.edu/data/web-Amazon.html

[12]From www.quora.com on AI, Arts, Books, CS, Economics, History, Home, Life Experience, Machine Learning, Nature, Science, Spirituality, Travel & Universe.

[13]Paired t-test performed on annotated HIT results with Amazon and BBC data sets partitioned into five parts (100 documents each).

| Data Set | Amazon | | | | BBC | | | |
|---|---|---|---|---|---|---|---|---|
| Approach | Prec@10 | Unamb@10 | Accuracy | p-val | Prec@10 | Unamb@10 | Accuracy | p-val |
| *classifier* | **0.437** | **0.604** | **0.604** | - | **0.589** | **0.885** | **0.778** | - |
| *classifier-no-text* | 0.371 | 0.518 | 0.545 | *** | 0.536 | 0.826 | 0.736 | *** |
| *classifier-no-text-no-cent* | 0.365 | 0.499 | 0.532 | *** | 0.495 | 0.771 | 0.701 | *** |
| *classifier-only-text* | 0.400 | 0.541 | 0.560 | ** | 0.538 | 0.823 | 0.736 | *** |
| *classifier-only-cent* | 0.337 | 0.457 | 0.500 | *** | 0.478 | 0.738 | 0.682 | *** |

Table I: Performance results from Amazon Mechanical Turk on Amazon and BBC data sets for the different approaches. (p-val indicates, for each competitor to *classifier*, whether the p-value from paired t-test is less than 0.005 (**) or 0.001 (***).)

| | NDCG | | |
|---|---|---|---|
| Approach | Macro | Micro | p-val |
| *classifier* | 0.343 | 0.351 | - |
| *classifier-only-text* | 0.306 | 0.314 | *** |
| *classifier-only-cent* | 0.262 | 0.270 | *** |
| *classifier-no-text* | 0.319 | 0.325 | ** |
| *classifier-no-text-no-cent* | 0.330 | 0.334 | |

Table II: NDCG results for Quora dataset. (p-val for t-test on Macro-NDCG was $< 0.005$ (**) or $< 0.001$ (***) compared to *classifier*.)

data and within each of the BBC categories. It is interesting to observe that text and graph features yield similar performance in isolation, but provide large gains when considered together.

Table II reports the NDCG results on Quora, and show that the proposed *classifier* module achieves highly statistically significant (paired t-test with 14 data partitions report p-value $< 0.001$) performance improvements compared to existing approaches, and demonstrates the scalability of our approach. Observe, that although *classifier* is trained only on 50 documents, it identifies good topic labels with high accuracy and exhibits robustness across various domains without re-training.

## VIII. Case Studies

In this section, we present two case studies to show the potential of ANNOTATE in topically organizing unstructured content for large-scale real applications.

### A. Indexing YouTube Videos

There is a large and ever-increasing body of multimedia content on the web and private repositories. Currently, time-consuming manual effort is required to organize and structure these videos through keywords provided by the content creator. Furthermore, these keywords are inconsistent, leading to poor organizational structure. Ideally, we would like to have indices for the multimedia content similar to the manually curated indices for the books. Such indices will give a quick glimpse of the different topics covered in the content, at different levels of granularity. Also, it will make it easy to navigate the content. For instance, in online course lectures, the students would like to know the topics and sub-topics covered in each lecture and be able to quickly navigate

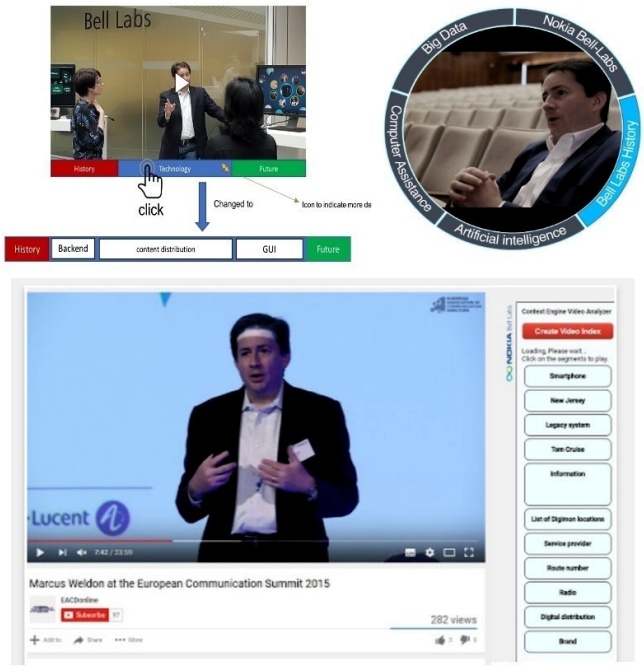



Figure 4: Chrome browser extension leveraging ANNOTATE for topically indexing YouTube videos

to a specific sub-topic. We have developed an extension of the popular Chrome browser for this purpose [14]. This publicly available extension leverages ANNOTATE to create video indices *automatically*. Figure 4 depicts an example video and shows the topic indices created by ANNOTATE . By looking at the indices, one can quickly understand that in this video, the speaker talked about smartphones, legacy telecommunication systems, service providers, radio access, digital distribution and branding issues. In addition, one also finds out that there was some quip about Tom Cruise as well.

The indexing works at different levels of granularity. As shown in the top-left of Figure 4, clicking on a topic can reveal the subtopics covered in the segment. These finer-granularity sub-topics facilitate the search for specific information.

This indexing enables smarter navigation of the multimedia content. As shown in the top-right of Figure 4, the top-

[14] chrome.google.com/webstore/detail/npkafmpocoljkekkbccdkacmjpacoljd

level indices can be overlaid on the video such that clicking on a topic label takes the viewer to the starting point of the corresponding segment.

Note that these indices also conceptually link a segment with segments of other videos through the layer of canonical topics. This is particularly handy when a viewer wants to find other content segments on the same topic. For instance, one may wish to find how different news sources covered a specific news item, given a collection of longer duration prime-time news videos.

The Chrome extension directly utilizes the YouTube APIs to access the subtitles/closed captions of the multimedia content. This enables the extension to get better text transcripts than relying on publicly and freely available speech-to-text systems. It then identifies the noun phrases from the text transcript using carefully defined rules. These noun phrases are then disambiguated to the Wikipedia category taxonomy using the PML disambiguation system [34]. Alternatively, the annotation and disambiguation from TagMe [33] can also be leveraged for this application.

Following the disambiguation, ANNOTATE is leveraged to segment the YouTube videos and then to assign the most relevant topic label to each segment. This enables the indexing functionalities described earlier.

### B. *ANNOTATE for tagging text content*

As described in Section VII, ANNOTATE finds highly accurate topic labels for longer text such as news as well as shorter text such as Quora answers. For applications like news classification, ANNOTATE can be trained to output general topic labels such as "U.K. Politics", "Cricket" etc. while for applications such as labeling Stack Overflow (a question answering platform where most questions are related to a narrow topic of Computer Science/Programming Languages), ANNOTATE can be trained to provide more specific topics such as "graph-layout in javascript."

ANNOTATE can also be used to tag social media posts from internal social networks of enterprises such as Yammer. In this case, the taxonomy used in ANNOTATE can either be a domain-specific taxonomy with concepts related to the enterprise' core business or it can be a general-purpose taxonomy extended with such concepts (e.g., using techniques in [4]).

Note that even for longer texts like news articles and publications, ANNOTATE does not need to segment the document as most text documents pertain to a singular topic. However in this case, ANNOTATE can provide multiple labels for the same text documents as the text document may cover diverse topics.

Figure 5 displays a screenshot of the ANNOTATE front-end for tagging multimodal content. For the text data, ANNOTATE provides API to deal with long formal texts as well as short informal text. In addition to the API, the ANNOTATE front-end also allows the text to be input in a free-form text box, which can be very useful to quickly assess the quality of the labels.

Figure 5: ANNOTATE for topically organizing unstructured multi-modal contents

## IX. Conclusion

We proposed a novel framework, ANNOTATE for automatic unstructured multi-modal content labeling with topics from Wikipedia taxonomy, by selecting a small set of efficiently computable directed graph and text features for learning to extract accurate labels characterizing the context spectrum. We showed that our model requires few training examples (easily obtainable in crowd-sourced manner) and is robust across diverse domains with no need for re-training. Experimental results on various real-life data showed our method to significantly outperform existing methods in terms of label accuracy.

## References

[1] O. Medelyan, I. H. Witten, and D. Milne, "Topic indexing with wikipedia," in *AAAI WikiAI workshop*, 2008, pp. 19–24.

[2] M. Grineva, M. Grinev, and D. Lizorkin, "Extracting key terms from noisy and multi-theme documents," in *WWW*, 2009, pp. 661–670.

[3] A. Hotho, S. Staab, and G. Stumme, "Ontologies improve text document clustering," in *ICDM*, 2003, pp. 541–544.

[4] N. Vedula, P. K. Nicholson, D. Ajwani, S. Dutta, A. Sala, and S. Parthasarathy, "Enriching taxonomies with functional domain knowledge," in *SIGIR*, 2018, pp. 745–754.

[5] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: a core of semantic knowledge unifying WordNet and Wikipedia," in *WWW*, 2007, pp. 697–706.

[6] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine Learning*, vol. 73, no. 2, p. 185, 2008.

[7] W. Bi and J. T. Kwok, "Multi-label classification on tree- and dag-structured hierarchies," in *ICML*, 2011, pp. 17–24.

[8] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, "Kernel-based learning of hierarchical multi-label classification models," *JMLR*, vol. 7, pp. 1601–1626, 2006.

[9] Z. Ren, M. H. Peetz, S. Liang, W. van Dolen, and M. de Rijke, "Hierarchical multi-label classification of social text streams," in *SIGIR*, 2014, pp. 213–222.

[10] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene, "Unsupervised graph-based topic labelling using dbpedia," in *WSDM*, 2013, pp. 465–474.

[11] E. Meij, W. Weerkamp, and M. de Rijke, "Adding semantics to microblog posts," in *WSDM*, 2012, pp. 563–572.

[12] S. Dumais and H. Chen, "Hierarchical classification of web content," in *SIGIR*, 2000, pp. 256–263.

[13] A. Aker, E. Kurtic, A. R. Balamurali, M. L. Paramita, E. Barker, M. Hepple, and R. J. Gaizauskas, "A graph-based approach to topic clustering for online comments to news," in *ECIR*, 2016, pp. 15–29.

[14] N. Aletras and M. Stevenson, "Labelling topics using unsupervised graph-based methods," in *ACL*, 2014, pp. 631–636.

[15] P. Treeratpituk and J. Callan, "Automatically labeling hierarchical clusters," in *DG.O*, 2006, pp. 167–176.

[16] J. H. Lau, D. Newman, S. Karimi, and T. Baldwin, "Best topic word selection for topic labelling," in *COLING*, 2010, pp. 605–613.

[17] M. Muhr, R. Kern, and M. Granitzer, "Analysis of structural relationships for hierarchical cluster labeling," in *SIGIR*, 2010, pp. 178–185.

[18] J. H. Lau, K. Grieser, D. Newman, and T. Baldwin, "Automatic labelling of topic models," in *ACL*, 2011, pp. 1536–1545.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[20] T. Hofmann, "Probabilistic latent semantic analysis," in *UAI*, 1999, pp. 289–296.

[21] Q. Mei, X. Shen, and C. Zhai, "Automatic labeling of multinomial topic models," in *KDD*, 2007, pp. 490–499.

[22] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing and Mining*, vol. 2007, pp. 1–13, 2007.

[23] W. Bi and J. T. Kwok, "Efficient multi-label classification with many labels," in *ICML*, 2013, pp. 405–413.

[24] G. Tsoumakas, I. Katakis, and I. Vlahavas, *Mining Multi-label Data*. Springer US, 2010, pp. 667–685.

[25] F. Tai and H. T. Lin, "Multi-label classification with principal label space transformation," *Neural Comput.*, vol. 24, no. 9, pp. 2508–2542, 2012.

[26] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[27] A. Bavelas, "Communication patterns in task-oriented groups," *JASA*, vol. 22, no. 6, pp. 725–730, 1950.

[28] S. Fortunato, V. Latora, and M. Marchiori, "Method to find community structures based on information centrality," *Physical Review*, vol. 70, no. 5, 2004.

[29] M. E. J. Newman, "A measure of betweenness centrality based on random walks," *Social networks*, vol. 27, no. 1, pp. 39–54, 2005.

[30] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *IJCAI*, 1995, pp. 448–453.

[31] D. Sánchez, M. Batet, and D. Isern, "Ontology-based information content computation," *Knowledge Based Systems*, vol. 24, no. 2, pp. 297–303, 2011.

[32] P. Wang, J. Hu, H. J. Zeng, and Z. Chen, "Using Wikipedia knowledge to improve text classification," *Knowledge and Information Systems*, vol. 19, no. 3, pp. 265–281, 2009.

[33] P. Ferragina and U. Scaiella, "TAGME: on-the-fly annotation of short text fragments (by wikipedia entities)," in *CIKM*, 2010, pp. 1625–1628.

[34] T. Mai, B. Shi, P. K. Nicholson, D. Ajwani, and A. Sala, "Scalable disambiguation system capturing individualities of mentions," in *LDK*, 2017, pp. 365–379.

[35] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia - A crystallization point for the web of data," *J. of Web Sem.*, vol. 7, no. 3, pp. 154–165, 2009.

[36] J. Lacasta, J. N. Iso, and F. J. Z. Soria, *Terminological Ontologies - Design, Management and Practical Applications*. Springer, 2010.

[37] O. Suominen and E. Hyvönen, "Improving the quality of SKOS vocabularies with skosify," in *EKAW*, 2012, pp. 383–397.

[38] O. Suominen and C. Mader, "Assessing and improving the quality of SKOS vocabularies," *J. Data Semantics*, vol. 3, no. 1, pp. 47–73, 2014.

[39] M. Fossati, D. Kontokostas, and J. Lehmann, "Unsupervised learning of an extensive and usable taxonomy for dbpedia," in *SEMANTICS*, 2015, pp. 177–184.

[40] J. Sun, D. Ajwani, P. K. Nicholson, A. Sala, and S. Parthasarathy, "Breaking cycles in noisy hierarchies," in *WebSci*, 2017, pp. 151–160.

[41] R. Nallapati, "Discriminative models for information retrieval," in *SIGIR*, 2004, pp. 64–71.

[42] P. Li, C. J. C. Burges, and Q. Wu, "Mcrank: Learning to rank using multiple classification and gradient boosting," in *NIPS*, 2007, pp. 897–904.

[43] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *ICML*, 2006, pp. 377–384.

[44] C. Yang and J. Wen, "Text Categorization Based on a Similarity Approach," in *ISKE*, 2007.