Learning Semantic Features from Web Services

Mário Antunes
Instituto de Telecomunicações
Universidade de Aveiro
Aveiro, Portugal
Email: mario.antunes@av.it.pt

Diogo Gomes
Instituto de Telecomunicações
Universidade de Aveiro
Aveiro, Portugal
Email: dgomes@av.it.pt

Rui Aguiar
Instituto de Telecomunicações
Universidade de Aveiro
Aveiro, Portugal
Email: ruilaa@av.it.pt

***Abstract*—In recent years the technological world has grown by incorporating billions of small sensing devices, collecting and sharing real-world information. As the number of such devices grows, it becomes increasingly difficult to manage all these new information sources. There is no uniform way to share, process and understand context information. It is our personal belief that IoT and M2M scenarios will only achieve their full potential when all the devices will work and learn together without human interaction. In this paper we review the most relevant semantic metrics and propose a new unsupervised model that minimizes sense-conflation problem. Our solution was evaluated against Miller-Charles dataset, outperforming our previous work in every metric.**

***Index Terms*—IoT, M2M, context information, semantic similarity**

I. Introduction

The Internet of Things (IoT) is a paradigm where everyday devices can be equipped with identifying, sensing and processing capabilities. This allows them to communicate with one another, other devices and even services on the Internet to accomplish some objective. A cornerstone to this connectivity landscape is machine-to-machine (M2M). M2M generally refers to information and communication technologies able to measure, deliver, digest and react upon information autonomously, i.e. with none or minimal human interaction.

Context-awareness is an intrinsic property of IoT and M2M scenarios. As discussed in [1] an entity's context can be used to provide added value: improve efficiency, optimize resources and detect anomalies. However, recent projects follow a vertical approach, devices/manufacturers share context information with a different structure, leading to information silos and low interoperability. This has hindered interoperability and the realisation of even more powerful IoT and M2M scenarios.

Context information is an enabler for further data analysis, potentially exploring the integration of an increasing number of information sources. The common definitions of context information [2], [3] are so broad that any information related to an entity can be considered context information. These definitions also do not provide any insight about the structure of context information. Currently there is no uniform way to share/manage vast amounts of M2M information.

It is possible (but unlikely) that in the future a context representation standard will be widely adopted. Until then, we have to deal with multiple context representations. In previous work we addressed this challenge, and proposed a novel d-dimension organization model [1].

IoT devices share a vast diversity of information, commonly in textual form. In a previous work we discussed the importance of semantic features and similarity for M2M scenarios [4]. We proposed an unsupervised method to learn distributional profiles from public web services. This method also allow us to organize, extract and cluster information based on concepts and not on sub-strings nor regular expressions. Apart from context-aware applications, several other areas benefit from semantic based context organization. For example these methods could provide a decisive contribution towards the exploration of name-based information centric network architectures in IoT environments [5]. Namely, the application of inference mechanisms into the content-reaching operations of the networking fabric itself can be used to have the network better mimic the complex relationships between devices (e.g., sensors, actuators), their generated content (e.g., temperature values with different units) and its dissemination towards interested entities.

Although our previous solution achieved a good score in the Miller-Charles dataset [6], we ended up proposing some improvements to our model. In this paper we discuss the implications of learning from noisy corpus and the impact of capturing multiple words senses in a single distributional profile. A reliable semantic metric should return the distance between the intended senses, which often tends to be the semantic distance between their closest senses. We developed a new unsupervised model that outperforms our previous solution and minimizes the issue of multiple word senses.

The remainder of this paper is organized as follows. In Section II we discuss semantic similarity and present the most relevant methods. We discuss our previous method and proposed several improvements in Section III. Section IV contains implementation details of our prototype. The results of our evaluation are in Section V. Finally, the discussion and conclusions are presented in Section VI.

II. Background and Related Work

Semantic distance/similarity is a property of lexical units, typically between words but this notion can be generalized to larger units such as phrases, sentences, etc. Two words are considered semantically close if there is a lexical semantic relation between them. There are two types of lexical relations: classical relation such as synonyms, antonyms and hypernymy and ad-hoc non-classical relation, such as cause-and-effect. If the closeness in meaning is due to a certain classical relation, then the terms are said to be semantically **similar**. On the other hand, semantic **relatedness** is the term used to describe the more general form of semantically closeness caused by any semantic relation. For instance the nouns *liquid* and *water* are both semantically similar and related, whereas the nouns *water* and *boat* are semantically related but not similar.

There are roughly three kinds of semantic measure: (1) lexical-resource-based measures that rely on manually created resources such as Wordnet; (2) corpus-based measures that rely only on co-occurrence statistics from large corpora; and (3) hybrid measures that are distributional in nature, and also exploit information from a lexical resource.

Lexical-resource-based measures rely on manually created and annotated lexical resources, such as WordNet[7], to determine the distance between two words. WordNet is a manually-created hierarchical network of nodes (taxonomy), where each node represents a fine-grained concept or word-sense. An edge between two nodes represents a lexical semantic relation such as hypernymy or troponymy. WordNet interlinks not just word forms (strings of letters) but specific senses of words. As a result, words that are found in proximity to one another in the network are semantically related. Several authors proposed semantic measures based on WordNet [8]–[10].

Semantic measures can only be used in languages that have (a sufficiently developed) WordNet. However, creating and maintaining lexical databases is a tedious task that requires human interaction. Furthermore, updating a lexical resource is expensive and there is usually a lag between the current state of language usage/comprehension and the resource representing it. For example, due to funding and staffing issues the WordNet project is no longer accepting comments and suggestions[1]. Due to these limitations, several authors proposed methods for large-scale acquisition of lexical knowledge, such as KnowNet [11] and BabelNet [12]. KnowNet is an extensible, large and accurate knowledge base, which as been derived by semantically disambiguating small portions of Topic Signatures [13] acquired from the Web. BabelNet is a very large, wide-coverage multilingual semantic network. It combines lexicographic and encyclopedic knowledge from WordNet and Wikipedia.

Besides these, several other methods exist to build large semantic networks. However, they rely on some sort of structured information, most of them maintained by human users. For example, BabelNet relies on WordNet and Wikipedia, while KnownNet relies on Topic Signatures. Although the information exchange in IoT/M2M scenarios is limited in vocabulary, usually consists of very specialized words associated with specific fields, topics and contexts. As a consequence, the lexical resource may not contain the correct vocabulary or even the relevant associations between the words.

Strictly corpus-based measures distributional similarity rely on the hypothesis that words that occur in similar contexts tend to be semantically close [14], [15]. The set of contexts of each target word u is represented by its distributional profile, in other words the set of words that tend to co-occur with u within a certain distance, along with numeric scores signifying this co-occurrence tendency with u. Measures such as cosine and α-skew divergence [16] are used to determine how close two distributional profiles are. These methods are very appealing because they rely solely on raw text, however they tend to perform poorly when compared with lexical-resource-based measures. See [17] for more information on distributional profiles.

The methods do not required a lexical-resource, but require a large corpus with representative usages of the target words. Due to the poor vocabulary present in M2M scenarios, the corpus made up from the information shared by M2M devices is not suitable to learn distributional profiles. Creating and maintaining large relevant corpus for M2M scenarios is a time consuming task that requires human interaction. The diversity of information and the poor vocabulary represent additional difficulties. Our previous solution [4] minimizes this issue using public web services to gather corpus and learn distributional profiles. It is important to mention that the primary objective of this work is to develop semantic features and metric that are usable in M2M scenarios. Devices in a M2M scenario may not have enough processing power or memory to analyze large corpus of raw text. We are trying to develop methods that extract reliable distributional profiles with the least amount of raw text.

Another important issue is sense-conflation problem. The distributional profile of a target word u conflates information about potentially many senses of u. Some authors [18] proposed hybrid measures that are distributional in nature but also rely on lexical resources to exploit the manually encoded information to overcome the sense-conflation problem. For example they extract distributional profiles for each sense of a word. They use categories from a Roget-style thesaurus [19]–[21] as coarse sense or concepts. A Roget-style thesaurus classifies all word types into approximately 1000 categories. Words with more than one sense are listed in more than one category. Each category has a head word that best represents the meaning of all the words in that category. The distance between words u and v is the closest distance between all their possible senses. Hybrid methods require a lexical resource, as such these

[1] http://wordnet.princeton.edu/wordnet/

methods have exactly the same disadvantages as lexical-resource-based measures for M2M scenarios. However, in this paper we proposed an unsupervised learning method to identify categories without the need of a Roget-style thesaurus.

It is worth mentioning that the previous solutions provide very accurate methods to estimate semantic similarity. However, those solutions rely heavily on structured information or well maintained corpus. The ever increasing number of IoT devices, M2M scenarios and applications makes it very difficult to build and maintain semantic networks or clean relevant corpus. The method we propose in this paper trades accuracy with flexibility and simplicity. Our solution does not require a specialized (large) corpus, and learns distributional profiles through Web Services using minimal textual information.

III. Distributional profiles from Public Web Services

Given a target word u we use public Web Services, namely search engines, to gather a potentially relevant corpus and extract the word u distributional profile. The profile is built based on proximity, which means if a word w is within the neighborhood of a target word u it is properly processed and extracted. This distributional profile of a word is defined as

$$DPW(u) = \{w_1, f(u, w_1); ...; w_n, f(u, w_n)\}$$

where u is the target word, w_i are words that occur with u and f stands for co-occurrence frequency (can be generalized for any strength of association metric). A distributional profile can also be interpret as a vector that represent a point in high dimensional space, each word w_i represent a dimension and $f(u, w_i)$ represents the value of the point in that dimension. From this point onward we will refer to words inside a DPW as dimensions. We evaluate the similarity between two DPW with cosine similarity:

$$cosine(u, v) = \frac{\sum_{i=1}^{n} f(u, w_i) \times f(v, w_i)}{\sqrt{\sum_{i=1}^{n} f(u, w_i)^2} \times \sqrt{\sum_{i=1}^{n} f(v, w_i)^2}}$$

Other similarity measures can be used, however cosine is invariant to scale. This similarity metric does not take into account the vector's magnitude, only their direction.

Although using public Web Services as a source has important advantages, it also has some disadvantages. Distributional profiles can be noisy, and contain several dimensions with low information. Also, a profile can contain several senses of the target word (sense-conflation). These issues decrease accuracy, and limit the potential of this method. In many practical scenarios we are interested in the semantic distance between the intended senses, which often tends to be the semantic distance between their closest senses.

We developed two filters to reduce unwanted dimensions on the DPW. The first filter uses stemming to merge words that have the same stem, eliminating issues with plural words. Our previous model used stem as dimensions and not words, as such it did not suffer from this issue. We require the original words (not only the stems) in the DPW in order to retrieve a corpus for the clustering process.

The second filter uses statistical significance to discard low value dimensions, it is based on the p-value statistical significance test. We defined the null hypothesis (H_0) as the dimension is generated randomly and the alternative hypothesis (H_a) as the dimension is relevant. The rationale is to compare the value of each dimension with a IID (Independent and Identically Distributed) model, where all the words that compose the profile have exactly the same probability of appearing. If the dimension's value is highly unlikely, then we discard the null hypothesis and assume that the dimension is relevant. Every time the DPW learning method finds the target word u, it extracts the corresponding neighborhood. We count the number of distinct words extracted from the neighborhood (named V) and the total number extracted words (named P). Assuming that each words has the same probability of appearing, the probability of a word appear exactly k times is express as follows

$$p(k) = \frac{\binom{P}{k} \times (V-1)^{P-k}}{V^P}$$

Using the previous expression we can compute the probability of a word appearing at least k times as follows

$$p(\geq k) = 1 - \sum_{i=1}^{k} \frac{\binom{P}{i} \times (V-1)^{P-i}}{V^P}$$

Using the previous expression we compute the probability for each dimension, if the result is greater that a predefined p the dimension is discarded[2].

These filters minimize the impact of low value dimensions, improve accuracy and processing speed. However, they do not minimize the effect of sense-conflation, where a distributional profile can learn dimensions from multiple word senses. In order to minimize this issue we propose using clustering on the distributional profile to identify categories/word senses. The rational is that dimensions belonging to the same category are closer to each other than word from other categories. Clustering method requires a distance metric in order to group similar elements. From this point we will discuss similarity metric, knowing that a similarity can be converted to a distance using the following expression

$$D(i, j) = 1 - S(i, j)$$

Since we are dealing with semantic similarity a natural solution is to use cosine similarity over each dimension's distributional profile. However, as stated previously, profiles extracted from Web Services may contain multiple senses of the target word. Alternatively we can use co-occurrence frequency as a similarity metric. This metric does not take

[2]In the evaluation we used $p = 0.01$

into account the neighbourhood of a target word, preventing the previous stated issue. Although co-occurrence itself does not represent semantic relatedness, it is a good estimation. In Section V we evaluate the performance of both metrics.

The clusters do not represent word senses from a Roget-style thesaurus. Which means that there is not a one-to-one relation between the clusters and a word sense in a thesaurus. Conceptually the clusters are similar to categories in latent semantic analysis, and may not have a correspondence to our human perception. Since a cluster may not represent a classical word sense, from this point onward we will refer to them as categories. One implication of this statement is that some clusters represent relevant categories, while others represent low relevant categories or even noise. Consider the following scenario, two target words u and v are not related, but end up with the same noisy category. This category will match and increase similarity value producing a false positive.

In order to minimize this issue our model incorporates an affinity between the target word and each category. The affinity is computed as the average similarity between the target word an all the cluster's elements. After computing all the affinity values, they are scale between $[0, 1]$ with the following expression

$$a' = \frac{a - min(a)}{max(a) - min(a)}$$

By incorporating affinity our model minimizes the effect of low relevant and noisy categories.

After the clustering process and computing the affinity of each cluster, the distributional profile of multiple words categories ($DPWC$) is extracted from the DPW and grouped according to the clusters obtained. The profile is defined as follows

$$DPWC(u) = \begin{Bmatrix} a_1; \{w_1, f(u_1, w_1); ...; w_n, f(u_1, w_n)\} \\ ... \\ a_n; \{w_1, f(u_c, w_1); ...; w_n, f(u_c, w_n)\} \end{Bmatrix}$$

where u is the target word, w_i are words that occur with u in a certain category, f stands for co-occurrence frequency and a_i is the affinity between u and a category. Finally, the similarity between two $DPWC$ is given by the following expression

$$sim(u, v) = max(cosine(u_c, v_c) \times (a_{u_c} + a_{u_v} \Big/ 2))$$

where u_c and v_c represent a specific category from u and v respectively and a represents the category's affinity . Our final similarity measure is the higher similarity between all the possible categories weighted by the average category's affinity.

IV. Implementation

In this section we discuss important details about our prototype implementation. Given a target word u our prototype uses web search engines to extract its $DPW(u)$ and $DPWC(u)$. Our prototype is divided into 5 different components as depicted in Figure 1. All the components were coded in Java.

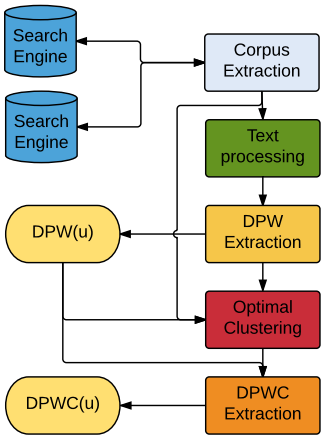

Fig. 1. Proposed DP extraction system's architecture.

The first component (corpus extraction) bridges our solution with web search engines. It can be used with any search engine, currently it uses two: Bing[3] and Faroo[4] This component basic function is to extract a corpus from search engines. The corpus is composed of snippets returned by searching for the target word. In a previous work [4] we compared the impact of using only snippets against the full webpages. We observed that snippets contain enough information to build reliable DPW. In this paper we are interested in comparing the performance of DPW against $DPWC$.

The second component (text processing) implements a pre-processing pipeline that cleans the corpus and divides it into tokens. First the snippets are tokenized and the resulting tokens are filtered using a stop word filter. Stop words are deemed irrelevant because they occur frequently in the language and provide little information. We used the MySQL stop word list[5]. For the exact same reason we also remove tokens that are too big or small. Any token with less than 3 or more than 12 (6 is average word length in English) characters were removed from the pipeline.

The DPW extraction component analyses the output of the pipeline and extracts the DPW of the target word u. This component also applies the filters mentioned in Section III that minimize the issue with low relevant dimensions. After extracting and optimizing the DPW, we use the corpus extraction component to gather a corpus for each individual dimension and apply the clustering process. We used K-means++ [22] to cluster the profile dimensions and identify the categories. K-means++ is a

[3]https://datamarket.azure.com/dataset/bing/searchweb
[4]http://www.faroo.com/hp/api/api.html
[5]https://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html

variant of the well known and widely used K-means that improves both speed and accuracy.

These algorithms have a drawback, they require the number of clusters *a priori*. Normally gap statistics [23] are used to identify the ideal number of clusters from a possible range However, this method requires generating reference features based on the elements to compare the clustering with a uniform sample. DPW are highly dimensional by nature, meaning that using this method is quite expensive. As an alternative we used the framework proposed in [24], it only requires the number of dimensions. The number of dimensions in the target u word can be used as an estimation.

Finally, the $DPWC$ component uses the DPW and the clusters to return the $DPWC(u)$ of the target word, this component also computes the affinity between the target word and each category.

V. Performance evaluation

We evaluate our model against Miller-Charles dataset [6], a dataset of 30 word-pairs rated by a group of 38 human subjects. Currently there is no word similarity database specific for M2M scenarios. Since in the scope of this work we did not have the opportunity to develop a specific dataset for M2M scenarios, we used a well known general propose dataset. We intend to address this issue in the future. The word pairs are rated on a scale from 0 (no similarity) to 4 (perfect synonymy).

Normally Pearson correlation is used to evaluate distance measure against the ground truth. Correlation between sets of data is a measure of how well they are related. The correlation r can range from -1 to 1. An r of -1 indicates a perfect negative linear relationship between variables, an r of 0 indicates no linear relationship between variables, finally and an r of 1 indicates a perfect positive linear relationship between variables. In short, the highest correlation indicates the most accurate solution.

One advantage of Pearson correlation is independent from scale and distance metric. The rationale is that even in different scales if the linear correlation between the ground truth and the similarity metric is high then the performance is also high. Our model uses unsupervised learning methods to identify categories and improve accuracy. However, the improvement may not be equal to each word pair in the dataset, damaging the linear correlation. As such, we also evaluate our model using mean squared error (MSE), a typical performance metric used in regression problems. It is worth mentioning that in order to used MSE metric we had to scale the dataset score.

Finally, we evaluated the performance of $DPW(u)$, $DPWC(u)$ with and without affinity for different neighborhood dimensions and two distinct clustering metrics: based on co-occurrence and cosine similarity. We tested our models on corpus formed from the top 150 snippets returned by our search engines: Bing and Faroo. It is worth mention that the results in this paper are not directly comparable with the previous work [4].

The results of the evaluation are listed in Table I and Table II. Based on the results we can conclude that independently from the method, as the neighborhood's size increases the performance decreases. The optimal neighborhood's size appear to be between 3 and 5. $DPWC$ with or without affinity outperforms the previous model (DPW). $DPWC$ without affinity achieves the highest performance based on MSE metric, while $DPWC$ with affinity achieves the highest performance based on Pearson correlation metric. This indicates that the affinity weighting improves accuracy and help maintain the linear correlation of the similarity metric.

TABLE I
Performance evaluation based on cosine distance metric

Methods	Neighborhood size 3		5		7	
	Pearson	MSE	Pearson	MSE	Pearson	MSE
DPW	0.43	0.31	0.44	0.29	0.41	0.29
$DPWC$	0.46	**0.21**	0.35	**0.23**	**0.42**	**0.24**
$DPWC_{Aff}$	**0.5**	0.23	**0.46**	0.27	0.36	0.3

TABLE II
Performance evaluation based on co-occurrence distance metric

Methods	Neighborhood size 3		5		7	
	Pearson	MSE	Pearson	MSE	Pearson	MSE
DPW	0.43	0.31	0.44	0.29	**0.41**	0.29
$DPWC$	0.49	**0.21**	0.41	**0.19**	0.34	**0.21**
$DPWC_{Aff}$	**0.55**	0.24	**0.54**	0.25	**0.41**	0.27

Finally, clustering based on co-occurrence tends to outperform clustering based on cosine similarity. Table III shows the average clusters identified in each evaluation. Clustering based on cosine similarity tends to produce smaller number of clusters, which means that each category tends to be bigger. This increases the probability of grouping relevant categories with noisy dimensions, decreasing performance overall. The smaller number of clusters were achieved using cosine similarity and a neighborhood of size 7. This explains why the $DPWC$ with affinity did not outperform the other methods. Since the number of clusters was so small, the affinity calculation was affected by noisy dimensions inside relevant clusters.

TABLE III
Average number of clusters

Distance Metric	Neighborhood size 3	5	7
Co-occurrence	5.50	6.47	5.37
Cosine	4.18	4.90	3.08

VI. Discussion and Conclusions

The number of sensing devices is increasing at a steady step. Each one of them generates massive amounts of information. However, each device/manufactures share context information with different structure, hindering interoperability in M2M scenarios.

In this paper we discussed the most relevant semantic similarity metric and the drawbacks of our previous solution. Distributional profiles extracted from Web Services may contain noisy dimensions and several senses of the target word (sense-conflation). These issues decrease accuracy, and limits the potential of this method. We proposed a new semantic model that minimizes these effects.

Our solution was evaluated against Miller-Charles dataset [6], outperforming our precious model in every metric. There is still room for improvement, hypernyms can be used to learn more abstract dimensions improving performance among other improvements. Nevertheless, our model was able to learn distributional profiles from a small corpus, achieving a relative high accuracy.

Acknowledgement

This work has been partially funded by research grant SFRH/BD/94270/2013.

References

[1] M. Antunes, D. Gomes, and R. L. Aguiar, "Scalable semantic aware context storage," *Future Gener. Comput. Syst.*, vol. 56, no. C, pp. 675–683, Mar. 2016.

[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proc. of the 1st international symposium on Handheld and Ubiquitous Computing*, 1999, pp. 304–307.

[3] T. Winograd, "Architectures for context," *Hum.-Comput. Interact.*, vol. 16, no. 2, pp. 401–419, December 2001.

[4] M. Antunes, D. Gomes, and R. L. Aguiar, "Semantic features for context organization," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, 2015, pp. 87–92.

[5] J. Quevedo, M. Antunes, D. Corujo, D. Gomes, and R. L. Aguiar, "On the application of contextual iot service discovery in information centric networks," *Computer Communications*, 2016.

[6] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1–28, 1991.

[7] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, November 1995.

[8] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ser. ACL '94, 1994, pp. 133–138.

[9] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'95, 1995.

[10] S. Banerjee and T. Pedersen, "An adapted lesk algorithm for word sense disambiguation using wordnet," in *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, ser. CICLing '02, 2002, pp. 136–145.

[11] M. Cuadros and G. Rigau, "Knownet: Building a large net of knowledge from the web," in *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, ser. COLING '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 161–168.

[12] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[13] C.-Y. Lin and E. Hovy, "The automated acquisition of topic signatures for text summarization," in *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, ser. COLING '00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 495–501.

[14] J. Firth, "A synopsis of linguistic theory 1935-55," *Transactions of the Philological Society*, 1957.

[15] Z. Harris, *Mathematical Structures of Language*. John Wiley and Son, 1968.

[16] L. Lee, "On the effectiveness of the skew divergence for statistical language analysis," in *Artificial Intelligence and Statistics*, 2001, pp. 65–72.

[17] G. Hirst and S. Mohammad, "Semantic distance measures with distributional profiles of coarse-grained concepts," in *Modeling, Learning, and Processing of Text Technological Data Structures*, 2012, pp. 61–79.

[18] Y. Marton, S. Mohammad, and P. Resnik, "Estimating semantic distance using soft semantic constraints in knowledge-source-corpus hybrid models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, ser. EMNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 775–783. [Online]. Available: http://dl.acm.org/citation.cfm?id=1699571.1699614

[19] P. Roget, *Roget's International Thesaurus, 1st edition*. New York: Cromwell, 1911.

[20] W. Hüllen, *A history of Roget's Thesaurus: Origins, development, and design*. Oxford University Press Oxford, 2004.

[21] M. Jarmasz and S. Szpakowicz, "Roget's thesaurus and semantic similarity," *arXiv preprint arXiv:1204.0245*, 2012.

[22] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[23] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[24] D. T. Pham, S. S. Dimov, and C. Nguyen, "Selection of k in k-means clustering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 219, no. 1, pp. 103–119, 2005.