

Understanding Large Persons' Networks through Semantic Categorization

Alessio Palmero Aprosio ^{a,*}, Sara Tonelli ^a, Stefano Menini ^{a,b} and Giovanni Moretti ^a

^a *Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy*

E-mail: [aprosio,satonelli,menini,moretti]@fbk.eu

^b *University of Trento, Via Sommarive 18, 38123 Trento, Italy*

Abstract. In this work, we describe a methodology to interpret large persons' networks extracted from text by classifying cliques using the DBpedia ontology. The approach and the challenges faced when building networks based on persons' co-occurrence are discussed in detail, especially the problem of mention normalisation and coreference resolution. The classification methodology that first starts from single nodes and then generalises to cliques is effective in terms of performance and is able to deal also with nodes that are not linked to Wikipedia. The gold standard manually developed for evaluation shows that groups of co-occurring entities share in most of the cases a category that can be automatically assigned. The outcome of this work may be of interest in a Big Data scenario to enhance the visualisation of large networks and to provide an additional semantic layer on top of cliques, so as to ease the comprehension of the network from a distance.

Keywords: Persons' networks, Semantic Linking, DBpedia ontology, Clique Classification

1. Introduction

In recent years, humanities scholars have faced the challenge of introducing information technologies in their daily research activity to gain new insight from historical sources, literary collections and other types of corpora, now available in digital format. However, in order to process large amounts of data and browse through the results in an intuitive way, new advanced tools are needed, specifically designed for researchers without a technical background. Especially scholars in the areas of social sciences or contemporary history need to interpret the content of an increasing flow of information (e.g. news, transcripts, political debates) in short time, in order to quickly grasp the content of large amounts of data and then select the most interesting sources.

One effective way to highlight semantic connections emerging from documents, while effectively summarising their content, is a network. In order to

analyse concepts and topics present in a corpus, several approaches have been successfully presented to model text corpora as networks, based on word co-occurrences, syntactic dependencies [33] or Latent Dirichlet Allocation [14]. While these approaches focus mainly on concepts, other information could be effectively modeled in the form of networks, i.e. *persons*. Indeed, persons' networks are the focus of several important research projects in the humanities, for instance Mapping the Republic of Letters¹, where connections between nodes have been manually encoded as metadata.

When we move to a Big Data scenario, however, where scholars need to manage large amounts of textual data, new challenges related to the creation of persons' networks need to be tackled. First of all, the process must be performed automatically. This involves taking some a priori choices related to the creation and the weight of nodes, and the setting of edges. Another issue is related to the readability of the resulting net-

*Corresponding author. E-mail: aprosio@fbk.eu

¹<http://republicofletters.stanford.edu/>

work. Indeed, networks extracted from large amounts of data can include thousands of nodes and edges. While several libraries have been released to display and navigate networks, an overview of the content of large networks is difficult to achieve.

In this work, we present an extensive study related to the extraction of persons' networks from large amounts of text, analysing the impact of persons' disambiguation and coreference resolution on the task. Besides, we present a methodology to exploit Semantic Linking in order to ease the readability of large persons' networks by adding a semantic layer on top of them. This layer includes categories automatically leveraged from DBpedia, which have been assigned to cliques of nodes. Through this process, interpretation of networks, so-called *distant reading* [25], is made easier.

This work is part of the ongoing ALCIDE project, whose goal is to develop a platform for advanced document processing to support humanities scholars in their daily research [26]. A first evaluation of the network extraction and visualisation functionality, involving 18 users in a demo session and a focus group, highlighted the need to improve the readability of the resulting network. Adding semantic categories was the top-ranked suggestion provided by the participants.

The article is structured as follows: in Section 2 we discuss past works related to our task, while in Section 3 we provide a description of the steps belonging to the proposed methodology. In Section 4, the experimental setup and the analysed corpus are detailed, while in Section 5 an evaluation of node and clique classification is provided and discussed. In Section 6 we provide details on how to obtain the implemented system and the dataset, and finally we draw some conclusions and discuss future work in Section 7.

2. Related work

This work lies at the intersection of different disciplines. It takes advantage from studies on graphs, in particular research on the proprieties of cliques, i.e. groups of nodes with all possible ties among themselves. Cliques have been extensively studied in relation to social networks, where they usually represent social circles or communities [11,15,21]. Although we use them to model co-occurrence in texts and not social relations, the assumption underlying this work is the same: the nodes belonging to the same clique share some common properties or categories, which we aim

at identifying automatically using the Linked Open Data.

This work relies also on past research analysing the impact of pre-processing, in particular coreference resolution and named entity disambiguation, on the extraction of networks from text. The work presented in [5] shows that anaphora and coreference resolution have both an impact on deduplicating nodes and adjusting weights in networks extracted from news. The authors recommend to apply both pre-processing steps in order to bring the network structure closer to the underlying social structure. This recommendation has been integrated in our processing pipeline. Impact of named entity disambiguation on networks extracted from e-mail interactions is analysed in [6]. The authors argue that disambiguation is a precondition for testing hypotheses, answering graph-theoretical and substantive questions about networks, and advancing network theories. We base our study on these premises, in which we introduce a mention normalisation step that collapses different person mentions onto the same node if they refer to the same entity.

The authors of [16] describe how BBC integrates data and links documents across entertainment and news domains by using Linked Open Data. Similarly, in [27] Reuters News articles are connected in an entity graph at document-level: people are represented as vertices, and two persons are connected if they co-occur in the same article. The authors investigate the importance of a person using various ranking algorithms, such as PageRank. In [13] a similar graph of people is created, showing that relations between individuals can be guessed also connecting entities at sentence-level, with high precision and recall.

In [17], the Semantic Web is used to get a representation of educational entities, in order to build self-organised learning networks and go beyond course and curriculum centric models. The *Trusty* algorithm [18] combines network analysis and Semantic Web to compute social trust in a group of users using a particular service on the Web.

3. Methodology

We propose and evaluate a methodology that takes a corpus in plain text as input and outputs a network, where each *node* corresponds to a person and an *edge* is set between two nodes if the two persons are co-occurring inside the same sentence. Within the network, *cliques*, i.e. maximum number of nodes who

have all possible ties present among themselves, are automatically labeled with a category extracted from DBpedia. In our case, cliques correspond to persons that tend to occur together in text, for which we assume that they share some commonalities, or took part to the same events. The goal of this process is to provide a comprehensive overview of the persons mentioned in large amounts of documents and show dependencies, overlaps, outliers and other features that would otherwise be hard to discern. An example network with two highlighted cliques is shown in Fig. 1, displaying a screenshot of the ALCIDE system.

The creation of persons' network from text can be designed to model different types of relations. In case of novels, networks can capture dialogue interactions and rely on the conversations between characters [8]. In case of e-mail corpora, [6], edges correspond to emails exchanged between sender and addressee. Each type of interaction must be recognised with an ad hoc approach, for instance using a tool that identifies direct speech in literary texts. However, since our goal is to rely on a general-purpose methodology that applies to a Big Data scenario, our approach to network creation is based on co-occurrence, similar to existing approaches to the creation of concept networks.

3.1. Pre-processing

The corpus is first processed with a pipeline of Natural Language Processing (NLP) tools. The goal is to detect persons' names in the documents and link them to DBpedia. We use PIKES [3], a suite integrating different NLP modules and optimized to process large amounts of data in short time. Specifically, we first identify persons' mentions in the documents (e.g. 'J. F. Kennedy', 'Lady Gaga', etc.) with the Stanford Named Entity Recogniser [9]. Then, the Stanford Deterministic Coreference Resolution System [20] is run to identify which expressions refer to the same entity in each document. For instance, the expressions 'J. F. Kennedy', 'J. F. K.', 'John Kennedy' and 'he' may all be connected in a coreferential chain because they all refer to the same person. Finally, DBpedia Spotlight [4] and the Wiki Machine [2] are both run to link the entities in the text with the corresponding DBpedia pages. In particular, we consider only links that overlap with the NER annotation and belong to the `Person` category. We combine the output of the two tools, since past works proved that this outperforms the performance of single linking systems [32]. In case of mismatch between the output of the two linking an-

notations, the confidence values (between 0 to 1, provided by both systems) are compared, and only the more confident result is considered.

While PIKES is a complex tool (time, CPU and memory consuming), it is applicable also to large amounts of data, as it can be optimized and parallelized to reduce the process time, for example by deactivating some modules such as coreference resolution. Therefore, precision, recall and execution time can be balanced by tuning the active modules and the parameters of the extraction.

3.2. Linking filters

In order to improve linking precision, two filters have been implemented and can be activated on demand. The first is applied to *highly ambiguous entities*, because it is very likely that they are linked to the wrong Wikipedia page, so it may be preferable to ignore them during the linking process. An entity should be ignored if the probability that it is linked to a Wikipedia page (calculated as described in [29]) is below a certain threshold. For instance, the word *Plato* can be linked to the philosopher, but also to an actress, *Dana Plato*, a racing driver, *Jason Plato*, or a South African politician, *Dan Plato*. However, the probability that *Plato* is linked to the philosopher page is 0.93, i.e. the link to the philosopher is probably always right. In some cases, thresholds are very low, especially for common combinations of name-surname. For example, *Bob Johnson* can be linked to 21 different Wikipedia pages, all of them having similar thresholds (0.16 for the ice hockey player, 0.13 for the musician, 0.09 for the pitcher, etc.). We manually checked some linking probabilities and set the threshold value to 0.2, so that if every possible page that can be linked to that set of tokens has a probability < 0.2 , the entity is not linked.

The second filter exploits the Linked Open Data paradigm to filter out entities that are certainly wrong, by forcing constraints on some entity properties (for example, inferred by time and/or location properties). In our case study (see Section 4) we removed all links referring to entities whose *birthDate* > 1943 , since the documents in the corpus were created in 1960 and persons mentioned there were likely to be born before 1943.

3.3. Network creation

The goal of this step is to take in input the information extracted through pre-processing and filtering and

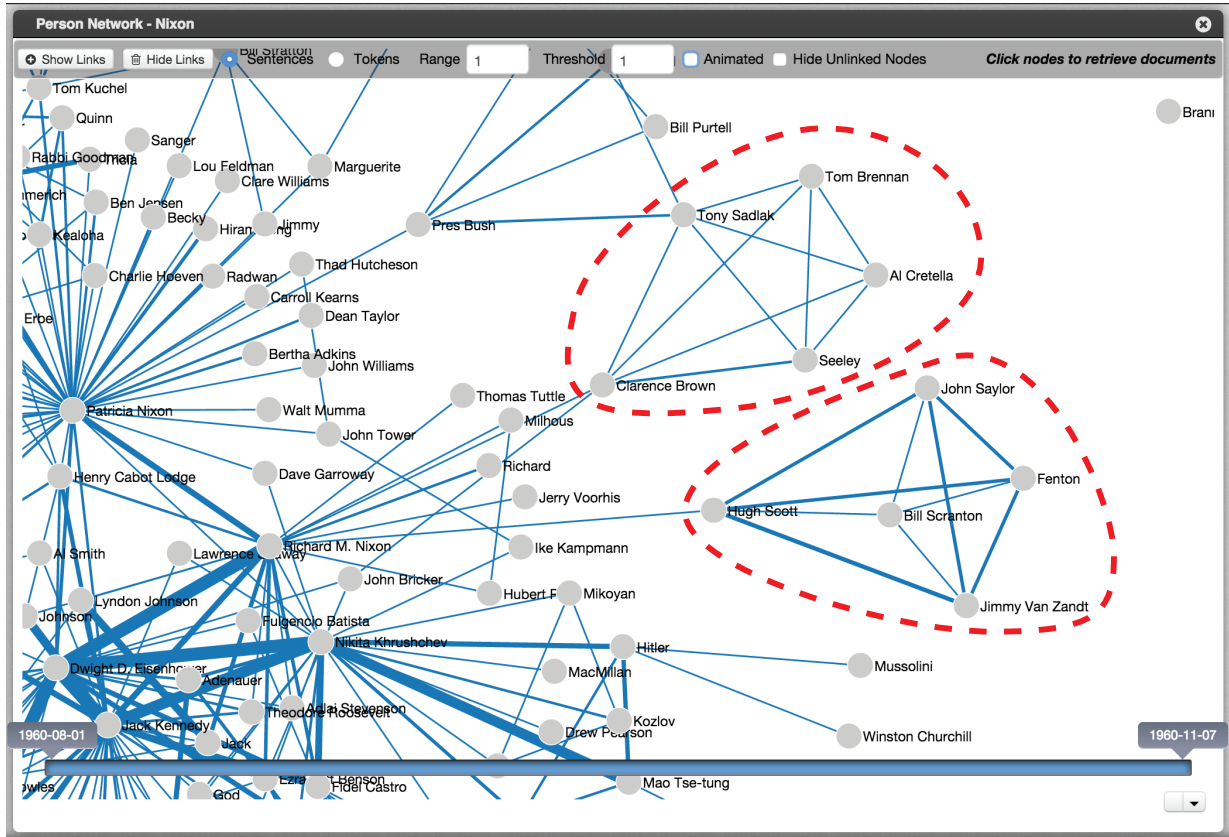


Fig. 1. Portion of network extracted with the ALCIDE tool from the corpus of Nixon's speeches. Cliques are marked in red.

produce a network representing person co-occurrences in the corpus. We assume that persons correspond to nodes and edges express co-occurrence. We build a person-person matrix where we assign an edge weight of 1 every time two persons are mentioned together in the same sentence². Every time a co-occurrence is repeated, the edge weight is increased by 1. The final output is a weighted undirected network where edge weights are co-occurrence frequency. Although the experiments presented in this article do not take into account such weights, they may be used in future to reduce the network size and select only the entities which are most mentioned.

A known issue in network creation is name disambiguation, i.e. identifying whether a set of person mentions refers to one or more real-world persons. This task can be very difficult because it implies understanding whether spellings of seemingly similar

names, such as 'Smith, John' and 'Smith, J.', represent the same person or not. The given problem can get more complicated, especially when people are named with diminutives (e.g. 'Nick' instead of 'Nicholas'), acronyms (e.g. 'J.F.K.') or inconsistently spelled.

In our approach, we consider two different steps to tackle the problem of disambiguation. In both cases, persons' information extracted at *document level* is processed and aggregated in order to obtain a global representation at *corpus level*.

Mention normalisation In this step, persons' mentions detected in each document are assigned to the network nodes based on a set of rules for mention normalisation and filtering. Entities comprising more than one token (i.e. complex entities) are collapsed onto the same node if they show a certain amount of common tokens (e.g. 'John F. Kennedy' and 'John Kennedy'). The approach is similar to the *first initial* method, that proved to reach 97% accuracy in past experiments[23]. However, we make the approach more robust because we deal also with simple entities (i.e. composed only of one token), spelling variants, etc. As for simple en-

²Even if the sentence window is arbitrary, it is common to consider this boundary also when manually annotating relations in benchmarks [24,13]

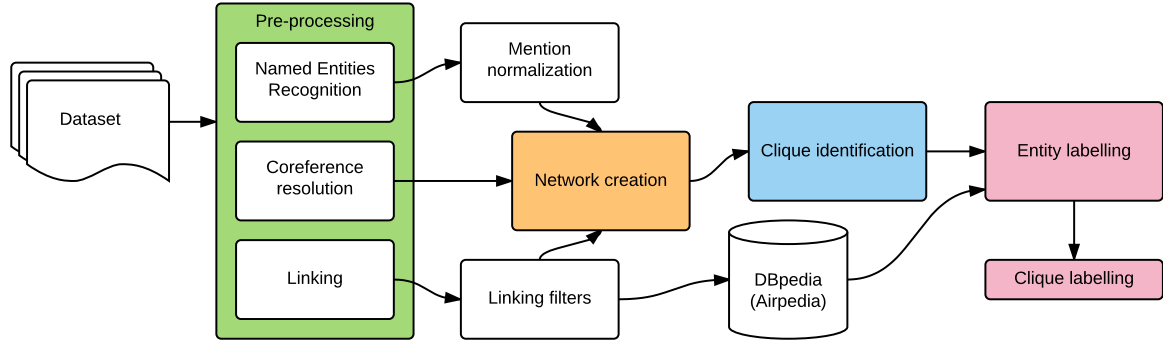


Fig. 2. Workflow of the whole system.

tities, they can be either proper names or surnames. To assess which simple entity belongs to which category, two lists of first and family names are extracted from biographies in Wikipedia, along with their frequency: a token is considered as a family name if it appears in the corresponding list and it does not appear in the first name list, unless it appears in the latter with a greater frequency. Tokens not classified as surnames are ignored and not included in the network. Tokens classified as surnames, instead, are merged with the node corresponding to the most frequent complex entity containing such surname. For example, the single mentions of ‘Kennedy’ are all collapsed onto the ‘John Fitzgerald Kennedy’ node, if it is more frequent in the corpus than any other node containing the same surname such as ‘Robert F. Kennedy’, ‘Ted Kennedy’, etc. After mention normalisation, the network has less nodes but it is more connected than the original version without normalisation (see Table 1).

Coreference resolution In this step, we integrate the output of the coreference resolution algorithm in the creation of the network, given that past experiments showed its usefulness in the extraction of persons’ networks from text [5]. Through coreference resolution, the number of edges is increased as well as the dimension of cliques. For instance, consider the following sentences, where expressions in italics are named entities and underlined tokens are annotations added by the coreference resolution system (numbers correspond to the entities found by the system):

“*John Kennedy*₁ contends that the Voice of America is weak, ignoring its growth under *President Eisenhower*₂. He₁ continues to assert that the President of the United States₂ could have expressed regrets or apologized to *Mr. Khrushchev*₃.”

After mention normalisation, *John Kennedy* is merged with the node of *John F. Kennedy* and *President Eisenhower* with the node of *Dwight D. Eisenhower*, and an edge is set between the two. The coreference resolution system would recognize an additional co-referring relation connecting *He* and *John Kennedy* as well as *The President of the United States* and *President Eisenhower*. This would lead to the creation of an edge also between the node of *John F. Kennedy* and that of *Nikita Khrushchev* and between *Nikita Khrushchev* and *Dwight D. Eisenhower*. In short, coreference resolution would add two connections to the *Nikita Khrushchev* node, which would be ignored if we applied only mention normalisation.

3.4. Clique identification and labelling

The last steps of the process include the identification of cliques, i.e. clusters of nodes with all possible ties among themselves (see Figure 1), and their classification by assigning a semantic category covering all nodes included in the clique.

In case of small datasets, existing algorithms can quickly find all maximal cliques inside a network (a maximal clique is a clique that cannot be enlarged by adding a vertex). The most efficient one is the Bron-Kerbosch clique detection algorithm [1]. Unfortunately, the algorithm takes exponential time $O(3^{n/3})$ (being n the number of vertices in the network), which means that it quickly becomes intractable when the size of the network increases. Since in our scenario we are not interested in listing *every* maximal clique, but we can instead limit the size of the cliques to a fixed value k (that can be arbitrary big, for example 10), the execution time drops to $O(n^k k^2)$, that is polynomial [7].

Clique labeling is performed according to the following algorithm. Let C be the set of cliques to be labeled. For each clique $c \in C$, let $c_i, i = (1 \dots k_c)$ be the nodes belonging to c (note that we extract cliques of different sizes, that's why we denote with k_c the size of the clique c). For each node c_i previously linked to a Wikipedia page (see Sections 3.1 and 3.2) we extract the corresponding DBpedia classes using Airpedia [30]. This system was chosen because it extends DBpedia coverage, classifying also pages that do not contain an infobox and exploiting cross-lingual links in Wikipedia. This results in a deeper and broader coverage of pages w.r.t DBpedia classes.

Let $\text{class}(c_i)$ be the set of DBpedia classes associated to an entity $c_i \in c$. Note that $\text{class}(c_i) = \emptyset$ for some c_i , as only around 50% of the entities can be successfully linked (see last column of Table 2).

For each clique, we define the first frequency function F' that maps each possible DBpedia class to the number of occurrences of that class in that clique. For example, the annotated clique

Gifford Pinchot \rightarrow Governor
 Theodore Roosevelt \rightarrow President
 Wendell Willkie \rightarrow [none]
 Franklin Roosevelt \rightarrow President

will result in

$$F'(\text{Governor}) = 1$$

$$F'(\text{President}) = 2.$$

As DBpedia classes are hierarchical, we compute the final frequency function F by adding to F' the ancestors for each class. In our example, as Governor and President are both children of Politician, F will result in

$$F(\text{Governor}) = 1$$

$$F(\text{President}) = 2$$

$$F(\text{Politician}) = 3.$$

Since in our task we focus on persons, we only deal with the classes dominated by Person (we ignore the Agent class, along with Person itself). Finally, we pick the class that has the highest frequency, and extend the annotation to the unknown entities. In the example, *Wendell Willkie* would be classified as Politician. The same class is also used to guess what the people in the clique have in common, i.e. a possible classification of the whole clique, to help the *distant reading* of the graph (see Section 1).

Table 1

Number of nodes and of cliques in the network with and without mention normalization (MN) and coreference resolution (COREF). In brackets, the average number of entities for each clique.

	w/o MN	MN
Number of nodes	4.754	4.261
Number of cliques		
w/o COREF	720 (4.62)	683 (4.60)
COREF	1.005 (4.91)	869 (4.80)

4. Experimental setup

We evaluate our approach on the corpus of political speeches uttered by Nixon and Kennedy during 1960 presidential campaign.³ The corpus contains around 1,650,000 tokens (830,000 by Nixon and 815,000 by Kennedy). Although these numbers are not typical of a Big Data scenario, our approach is designed to easily scale up and tackles one of the issues of working with Big Data, i.e. semantic interpretation and readability of processing results. Besides, the corpus is representative of political discourse, a domain in which large amounts of data are typically available and used by humanities scholars.

The full processing of the dataset on a single machine (12GB of RAM, 20 threads) took 170 minutes, at an average speed of about 162 tokens/s. Without the coreference resolution step, the processing time would decrease by fifty percent.

The corpus is first pre-processed as described in Section 3.1. Then, the recognized entities are linked and mention normalisation (MN) and coreference resolution (COREF) are performed, as a preliminary step to the creation of the network and the extraction of cliques (see Section 3.4). We show in Table 1 the impact of these two processes on the network dimension and on the number of extracted cliques.

Clique identification is performed by applying the Bron-Kerbosch clique detection algorithm (see Section 3.4), using the implementation available in the JGraphT package.⁴ After this extraction, we only work on cliques having at least 3 nodes, as smaller cliques would be too trivial.

Mention normalization reduces the number of nodes by 10% because it collapses different mentions onto

³The transcription of the speeches are available online by John T. Woolley and Gerhard Peters, The American Presidency Project (http://www.presidency.ucsb.edu/1960_election.php)

⁴<http://jgrapht.org/>

the same node. Consequently, the number of cliques decreases (see Table 1). Coreference resolution, instead, does not have any impact on the network dimension, but it increases the number of edges connecting nodes, resulting in an increment of the number of cliques and also of their dimension. The evaluation presented in the remainder of this article is based on a system configuration including both mention normalisation and coreference resolution.

5. Evaluation

Since the goal of this work is to present and evaluate a methodology to assign categories to cliques and make large persons' networks more readable, we first create a gold standard with two annotated layers, one at *node* and one at *clique* level. This data set includes 50 cliques randomly extracted from the clique list (see Section 3.4).

First, each node in the clique is manually annotated with one or more classes from the DBpedia ontology [19] expressing the social role of the person under consideration. For example, *Henry Clay* is annotated both as *Senator* and *Congressman*. For many political roles, the ontology does not contain any class (for instance, *Secretary*). In that case the person is labeled with the closest more generic class (e.g. *Politician*).

Then, for each clique, we identify the most specific class (or classes) of the ontology including every member of the group. The shared class is used as label to define the category of the clique. For example, a clique can be annotated as follows:

John Swainson → Governor
 G. Mennen Williams → Governor
 Thaddeus Machrowicz → Congressman
 Jim O'Hara → Congressman
 Pat McNamara → Senator
 [whole clique] → Politician.

In case no category covering all nodes exists, the *Person* class is assigned. For instance, a clique containing 3 nodes labeled as *Journalist* and 2 nodes as *President* is assigned the *Person* class.

The gold standard contains overall 204 persons grouped into 50 cliques, only 6 of which are labelled with the *Person* category. This confirms our initial hypothesis that nodes sharing the same clique (i.e. persons that tend to be mentioned together in text)

show a high degree of commonality. All entities in the gold standard are assigned at least one category. Since this task is performed by looking directly at the DBpedia ontology, also persons that are not present in Wikipedia are manually labeled. In case a node is ambiguous (e.g. six persons named *Pat McNamara* are listed in Wikipedia), the annotator looks at the textual context(s) in which the clique occurs to disambiguate the entity.

In Table 2, we report different stages of the evaluation performed by comparing the system output with the gold standard. We first evaluate the classification of the single nodes (*'node classification'*) by comparing the category assigned through linking with DBpedia Spotlight and the Wiki Machine to the class labels in the gold standard. Since our methodology assigns a category to a clique even if not all nodes are linked to a Wikipedia page, we evaluate also the effect of inheriting the clique class at node level (see row *'Extending to non-linked entities'*). Besides, we assess the impact of *'highly ambiguous entities'* on node classification, and the effect of removing them from the nodes to be linked (*'without highly ambiguous entities'*). For instance, we removed from the data the node of *'Bob Johnson'*, which may refer to 21 different persons (see section 3.4 for details). The last line in Table 2 shows the performance of the system on guessing the shared class for the entire clique.

For each entity that needs to be classified, the evaluation is performed as proposed by [22] for a similar hierarchical categorisation task. Figure 3 shows an example of the evaluation. The system tries to classify the entity *Dante Fascell* and maps it to the ontology class *Governor*, while the correct classification is *Congressman*. The missing class (question mark) counts as a false negative (*fn*), the wrong class (cross) counts as a false positive (*fp*), and the correct class (tick) counts as a true positive (*tp*). As in this task we classify only people, we do not consider the true positives associated to the *Person* and *Agent* classes.⁵ In the example above, classification of *Dante Fascell* influences the global rates by adding 1 *tp*, 1 *fn* and 1 *fp*. Once all rates are collected for each classification, precision (*p*), recall (*r*) and F_1 are calculated as follows:

⁵See <http://mappings.dbpedia.org/server/ontology/classes/> for a hierarchical representation of the DBpedia ontology classes.

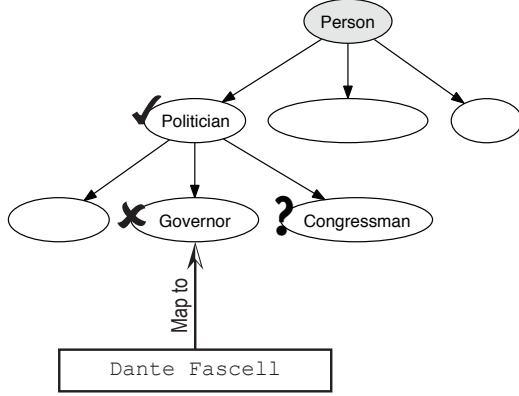


Fig. 3. Description of the evaluation.

$$p = \frac{tp}{tp + fp} \quad r = \frac{tp}{tp + fn} \quad F_1 = 2 \cdot \frac{p \cdot r}{p + r}.$$

Results in Table 2 show that the performance of node classification suffers from missing links, depending on the incomplete coverage of DBpedia Spotlight and the Wiki Machine, but also on the fact that some entities are not present in Wikipedia. However, this configuration achieves a good precision. In terms of F_1 , extending the class assigned to the clique also to non-linked entities yields a performance improvement, due to better recall.

Removing highly ambiguous entities is extremely beneficial because it boosts precision as expected, especially in combination with the strategy to extend the clique class to all underlying nodes. The setting based on this combination is the best performing one, achieving an improvement with respect to basic node classification both in precision and in recall.

Based on the best performing setting for node classification, we also evaluated the resulting clique classification, with the goal of assigning a category to clusters of interconnected nodes and easing the network comprehension. Results show that the task achieves good results and, even if not directly comparable, classification performance is higher than on single nodes. This shows that: (i) the DBpedia ontology is a resource suitable for this kind of task, where a generalisation step from nodes to cliques is needed; (ii) cliques tend to have a common category, meaning that persons co-occurring in texts show a high degree of commonality; and (iii) clique classification is an effective approach to discover the category of entities that are not present in Wikipedia.

6. Dataset and tool

The tool performing the workflow described in this paper is written in Java and released on GitHub⁶ under the GPL license, version 3. In the same repository one can find both the gold standard and the extracted dataset of people classified using the DBpedia ontology.

The documents used in our case study (Nixon and Kennedy discourses in the 1960 US Presidency election) are released under the NARA public domain license and may be reproduced. On our GitHub page one can find the dataset of the original Nixon and Kennedy speech transcriptions, along with the linguistic annotations applied in the pre-processing step (in NAF format [10], see Section 3.1).

7. Conclusions and Future Work

In this work, we presented an approach to extract persons' networks from large amounts of textual data based on co-occurrence relations. Then, we introduced a methodology to identify cliques and assigned them a category based on DBpedia ontology. This additional information layer is meant to ease the interpretation of networks, especially when they are particularly large.

We discussed in detail several issues related to the task. First of all, dealing with textual data is challenging because persons' mentions can be variable or inconsistent, and the proposed approach must be robust enough to tackle this problem. We rely on a well known tool for coreference resolution and we perform mention normalisation so that all mentions referring to the same entity are recognised and assigned to the same node. We also introduced two filtering strategies, one to deal with highly ambiguous entities and one to put temporal constraints on the recognised entities, in order improve the network quality.

Finally, we presented and evaluated a strategy to assign a category to the nodes in a clique and then, by generalisation, to the whole clique. The approach yields good results, especially at clique level, and is able to classify also entities that are not present in Wikipedia. The data manually annotated for the gold standard confirmed the initial hypothesis that co-occurrence networks based on persons' mentions can provide an interesting representation of the content of a document collection, and that cliques can effectively

⁶<https://github.com/dkmfbk/cliques>

Table 2
Evaluation of node and clique classification

Experiment	P	R	F_1	Classified entities
Node classification	0.689	0.481	0.566	139/204
Extending to non-linked entities	0.617	0.578	0.597	204/204
Node classification, without highly ambiguous entities	0.855	0.439	0.580	101/204
Extending to non-linked entities, without highly ambiguous entities	0.729	0.643	0.684	204/204
Clique classification	0.655	0.776	0.710	

capture commonalities among co-occurring persons. To our knowledge, this hypothesis was never proved before, and the clique classification task based on DBpedia ontology is an original contribution of this work.

In the future, we first plan to integrate the layer to visualise clique categories in the ALCIDE tool (Fig. 1), so that it can be exploited in a real research scenario. We are also going to further improve and extend nodes and cliques classification, for instance by applying clique percolation [28], a method used in Social Media analysis to discover relations between communities [12]. In our setting, it can be used to infer the DBpedia class of cliques that are not linked to Wikipedia. Almost-cliques have also been investigated in past works [31] and should be integrated in our system to increment the coverage of our approach by including more entities.

References

- [1] Coen Bron and Joep Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, September 1973.
- [2] Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. *Supporting Natural Language Processing with Background Knowledge: Coreference Resolution Case*, chapter 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7–11, 2010, Revised Selected Papers, Part I, pages 80–95. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [3] Francesco Corcoglioniti, Marco Rospoher, and Alessio Palmero Aprosio. A 2-phase frame-based knowledge extraction framework. In *Proc. of ACM Symposium on Applied Computing (SAC'16)*, 2016.
- [4] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.
- [5] J. Diesner and K.M. Carley. He says, she says. pat says, tricia says. how much reference resolution matters for entity extraction, relation extraction, and social network analysis. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, pages 1–8, July 2009.
- [6] Jana Diesner, Craig S. Evans, and Jinseok Kim. Impact of entity disambiguation errors on social network properties. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26–29, 2015*, pages 81–90, 2015.
- [7] Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science*, 141(1):109–131, 1995.
- [8] David K. Elson, Nicholas Dames, and Kathleen R. McKeown. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*, pages 138–147, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [9] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs Sampling. In *Proceedings of ACL '05*, pages 363–370. Association for Computational Linguistics, 2005.
- [10] Antske Fokkens, Aitor Soroa, Zuhaitz Beloki, Niels Ockeloen, German Rigau, Willem Robert van Hage, and Piek Vossen. Naf and gaf: Linking linguistic annotations. In *Proceedings 10th Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation*, pages 9–16, 2014.
- [11] Przemyslaw A. Grabowicz, Luca Maria Aiello, Victor M. Eguiluz, and Alejandro Jaimes. Distinguishing topical and social groups based on common identity and bond theory. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM '13*, pages 627–636, New York, NY, USA, 2013. ACM.
- [12] Enrico Gregori, L. Lenzini, and C. Orsini. k-clique communities in the internet as-level topology graph. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pages 134–139, June 2011.
- [13] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, ACL '04*, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
- [14] Keith Henderson and Tina Eliassi-Rad. Applying latent dirichlet allocation to group discovery in large graphs. In *Proceedings of the 2009 ACM Symposium on Applied Computing, SAC '09*, pages 1456–1461, New York, NY, USA, 2009. ACM.
- [15] Long Jin, Yang Chen, Tianyi Wang, Pan Hui, and A.V. Vasilakos. Understanding user behavior in online social networks: a survey. *Communications Magazine, IEEE*, 51(9):144–150, September 2013.
- [16] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections. In *The Semantic Web: Research and Applications: 6th European Semantic Web Conference, ESWC 2009 Heraklion, Crete, Greece*,

- May 31–June 4, 2009 Proceedings, pages 723–737, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [17] Rob Koper. Use of the semantic web to solve some basic problems in education: Increase flexible, distributed lifelong learning; decrease teacher's workload. *Journal of Interactive Media in Education*, 2004(1):Art–5, 2010.
 - [18] Ugur Kuter and Jennifer Golbeck. Semantic Web Service Composition in Social Environments. In *The Semantic Web - ISWC 2009: 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings*, pages 344–358, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
 - [19] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 2014.
 - [20] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.
 - [21] Julian McAuley and Jure Leskovec. Discovering social circles in ego networks. *ACM Trans. Knowl. Discov. Data*, 8(1):4:1–4:28, February 2014.
 - [22] I. Dan Melamed and Philip Resnik. Tagger evaluation given hierarchical tag sets. *Computers and the Humanities*, pages 79–84, 2000.
 - [23] Staša Milojević. Accuracy of simple, initials-based methods for author name disambiguation. *Journal of Informetrics*, 7(4):767 – 773, 2013.
 - [24] Alexis Mitchell, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstain, Lisa Ferro, and Beth Sundheim. *ACE-2 Version 1.0. LDC2003T11*. Linguistic Data Consortium, 2002.
 - [25] Franco Moretti. *Distant Reading*. Verso, London, 2013.
 - [26] Giovanni Moretti, Sara Tonelli, Stefano Menini, and Rachele Sprugnoli. ALCIDE: An online platform for the Analysis of Language and Content In a Digital Environment. In *Proceedings of the 1st Italian Conference of Computational Linguistics*, Pisa, 2014.
 - [27] Arzucan Özgür, Burak Cetin, and Haluk Bingol. Co-occurrence Network of Reuters News. *International Journal of Modern Physics C*, 19(05):689–702, 2008.
 - [28] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
 - [29] Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Automatic expansion of DBpedia exploiting Wikipedia cross-language information. In *Proceedings of the 10th Extended Semantic Web Conference*, 2013.
 - [30] Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Automatic Mapping of Wikipedia Templates for Fast Deployment of Localised DBpedia Datasets. In *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*, 2013.
 - [31] Jian Pei, Daxin Jiang, and Aidong Zhang. On mining cross-graph quasi-cliques. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 228–238, New York, NY, USA, 2005. ACM.
 - [32] Giuseppe Rizzo and Raphaël Troncy. NERD: A framework for unifying named entity recognition and disambiguation web extraction tools, 04 2012.
 - [33] Saatviga Sudhahar, Giuseppe A Veltri, and Nello Cristianini. Automated analysis of the us presidential elections using big data and network analysis. *Big Data & Society*, 2(1), 2015.