Tell me who are your friends, and I'll tell you who you are

Pablo Torres-Tramón, Conor Hayes

Insight Centre for Data Analytics, National University of Ireland Galway, Ireland
{pablo.torres,conor.hayes}@insight-centre.org

Abstract. Mentions of politicians in news articles can reflect politician interactions on their daily activities. In this work, we present a mathematical model to represent such interactions as a graph, and we use it to predict the political affiliation for a list of deputies of the Republic of Ireland. Our results show that there is a relation between political affiliations and mentions in news reports.

1 Background

An old Spanish proverb says that: "Tell me who are your friends, and I'll tell you who you are". A bunch of similar expressions are found in English, for instance: "Birds of a feather flock together". Though the former expression has been popularly attributed to Confucius, it can be found in several languages across the globe. In many cultures, human beings have built large societies based on the integration of diverse communities in one single body. Such an integration process involves the transmissions and assimilation of a set of features that eventually, will be used as a descriptors for that community. In other words, members of a community assimilate a set of common features that are generated within the community, in such way that it is possible to describe each individual using this common set.

The advent of news media and the new outcomes of communications has made possible to record and archive the interactions of large societies in a scale without precedent in human history. Every day, large volumes of data are created in the web, and stored as digital documents that encompass several stories such as political discussions, public opinions and personal thoughts. A prominent source of information, among those created and stored every day in the web, are news reports. In these documents, the interactions of relevant social actors, as well as the continuous political discussion, are constantly reported and evaluated by journalists and opinion leaders. News report archives are a good source for exploring and identifying behaviours for both, individual actors as well as the society. For instance, when a new bill is introduced to Parliament, a wide variety of actors will express their opinion in newspapers. Or, in case of economic struggles, opinion leaders will present their solution and defend it from news articles.

This characteristic of news articles makes them an ideal source of information in order to identify communities. In this paper, we introduce a graph model for

news reports and we use it for identifying well-established communities. Our intuition is that a news report reflects the opinion of a limited set of actors. The shared features among those actors can be used for identifying communities.

2 Problem statement

The narrative found in a news article describes a series of social interactions, including but not limited to: political discussion, legal procedures and opinion leaders. A regular reader can, for instance, identify social ties among politicians or identify trends in public opinion about an important matter. Assuming that reports of social actors interacting with their counterparts on their daily duties are included in newspapers constantly with some degree of confidence, we can develop a mathematical model that uses the news reports as input and returns a network containing the interactions of the actors. In particular, we state that if two actors are mentioned together in a news article, then it is highly probable that those actors are related to some degree. Such a degree, namely *relatedness*, indicates the affinity between two actors. We estimate relatedness directly from the number of ties between two actors, since this number implies that both actors are constantly mentioned together and, therefore, their relatedness must be higher.

Supposing that politicians have preferences in their duties, or they have biases toward particular actions, then news media reports should also reveal such preferences by reflecting a major number of such actives for any given politician. Of course, news media have biases as well, however we do not consider them in order to keep our model as simple as possible. Considering a finite set news reports and politicians for a fixed time period *a*) Do politician interactions in news reports follow an order or structure? *b*) If such an order exists, is it correlated with the government-parties structure or it correlated with a hidden structure?.

We believe that the interactions of politicians in news reports are highly correlated with the government-parties structure and their own individual preferences. Such correlations can be seen by using the a graph model to predict political affiliations. We assume that most news articles cover a single topic in one report and that they are related to the same socio-cultural background.

3 Proposed Method

In this section, we describe the graph model developed and the approach for predicting political affiliations.

3.1 Interaction Graph

Interactions are usually presented in written English as an interaction-triple T composed by a subject s, a predicate p and an object o. The subject is the generator of an action which is designated by the predicate. Usually a subject is

composed by a proper noun and a predicate by a verb. The predicate generally implies the existence of an object to which the action has an impact, however is not always the case. Some predicates do not have an object. A triple T can be written in English in different order, not necessary as s,p and o. Fortunately, the meaning of a triple when the order has changed is, in most of the case, the same.

Definition 1. *An interaction-triple is a tuple $T = (s, p, o)$ where $s \in S, p \in P$, $o \in \Phi \cup O$ and S, P, O are non-empty finite sets of subjects, predicates and objects respectively.*

We limited the set of subjects S and objects O to proper nouns only, since other nouns may not be relevant for determining communities. Moreover, we restricted proper nouns to Named Entities (N). Since both sets are composed of named entities ($S \subseteq N$, $O \subseteq N$), then the intersection may not be empty necessarily $|S \cap O| \geq 0$. That is, a named entity e_1 can be the subject for a triple T_1, and, at the same time, be the object for triple T_2. The resulting linking of interaction triples can be represented as a graph.

Definition 2. *Given a sample of interaction-triples $I = \{T_1, \ldots, T_n\}$, an undirected graph $G = (V, E)$ is an interaction graph such that $V = \bigcup_{T_i \in I}(\{s_i, o_i\})$ and $E = \{(s, o) \in V^2 | \exists T = (s, p, o) \in I\}$*

An interaction graph allows us to explore a written document according to the interactions represented in the text for different actors. However, the large variety of predicates affects the complexity of the analysis since it requires to identify the action involved and the respective subject and object. In order to simplify the transformation from text to interaction-graph, we restricted our predicate only to co-occurrence of named entities in the text. We called such co-occurrence a mention, as it follows:

Definition 3. *Given a chunk of text whose named entities are $N = \{n_1 \ldots, n_m\}$, we define a mention set as $I = \{(n_i, n_j) \in N^2 | \forall n_i, n_{j_{i<j}}\}$.*

Our attempt is to compare an interaction graph with the government-parties structure in order to predict if a given named entity is related to a government-parties element. When creating an interaction-graph, a set of named entities N must be identified first in each chunk of text. Then, we combine every set of named entities into one single set. The named entities are linked according to definition 3. The resulting graph represents the mentions of named entities in a corpus.

The input texts correspond to a collection of news articles. For example, let us consider the following paragraph from a news article: "Stephen Donnelly, one of the founders and co-leaders of the Social Democrats, has left the party in bitter circumstances just over a year after the party was established ... along with Roisn Shortall and Catherine Murphy, who were also his co-leaders[1]". We

[1] http://www.irishtimes.com/news/politics/stephen-donnelly-leaves-social-democrats-says-some-partnerships-simply-don-t-work-1.2780073

found the following entities: "Stephen Donnelly", "Social Democrats", "Roisín Shortall", "Catherine Murphy". Some of the co-occurrences triples for this report are the followings:

- $T_1 = (s_1 =$"Stephen Donnelly", $p_1 =$ co-occurr, $o_1 =$ "Social Democrats")
- $T_2 = (s_2 =$"Stephen Donnelly", $p_2 =$ co-occurr, $o_2 =$ "Roisín Shortall")
- $T_3 = (s_3 =$"Stephen Donnelly", $p_3 =$ co-occurr, $o_3 =$ "Catherine Murphy")
- $T_4 = (s_4 =$"Roisín Shortall", $p_4 =$ co-occurr, $o_4 =$ "Catherine Murphy")

Before adding news articles to our collection, we will filter them according to some fixed rules (they are explained in section 4.1). We require a filtering step to avoid the inclusion of a wide variety of topics in the final collection. Since we have an interest on the political interactions, the corpus must be composed mainly by news regarding politics. Therefore, other areas in a newspaper, such *sports* or *culture*, are considered noise and must be avoided. Filtering not only reduce the amount of noise in the collection but also limit the quantity of named entities in the graph and thus, it simplifies community finding algorithms. It is important to emphasize that a filtering process must be designed for a particular news portal. A generic filtering may decrease the performance of this step.

3.2 Named Entities

Several third-party libraries can be employed for determining named entities from an input text. Most of such libraries make use of Natural Language Processing (NLP) techniques. This task is generally known as Named Entity Recognition (NER). Although, most of the available software effectively find named entities, we are also interested in the nature of them, i.e. if they are referring to a person, organisation, location or other. For example, the nature of "Argentina" and "Germany" is *country*, while for "Apple Inc." and "Google" is *organisation*. This property, namely as *named entity type*, is essential for discriminating name entities, becoming easy to create filters based purely on named entity types. Since our objective is to predict the affiliation of politics, we are interested much more on named entities that are referring to people rather than other types such as organisation or location. A collection of named entities restricted to a particular type makes the interaction-triple graph homogeneous, less complex and simple to analyse.

Keeping only named entities referring to a person has two important implications. In first place, (as we stated in the previous paragraph) it simplifies the graph analysis since all named entities now have the same type. On the other hand, it removes key features of human interactions. For instance, when a place is mentioned in a news article, our graph will not include it, so the locality aspect of the interaction is about be missed. Making interactions de-localised creates a loss of information that, eventually, can generate a bias in the graph. We can reduce such a bias by analysing news articles that are highly associated with a particular region and cultural context. By similar arguments, reducing the temporal space of the articles avoids biases towards a unique time-frame. For such reason, we analyse articles whose times of creation are as close as possible.

Finally, named entities selected are only those whose type is is Person. Note that we have not specified the length of a chunk of text. Such length depends on the objective of the graph, and therefore is arbitrary to such objective. For instance, if a chunk is defined as a single sentence, then it is highly probable that the resulting graph will be sparse (very few nodes are linked) since the number of co-occurrence will be less in a single sentence than in a larger chunk. Instead, as a chuck becomes larger, graph would be more connected and eventually can become into a complete graph. In our case, each news article was used as a single chunk, therefore, the selection of named entities will take place at this level, and the co-occurrence will reflect the mentions of named entities in a single article.

3.3 Party Prediction

Most politicians in the western democracies belong to a political party. A political party is an organisation that groups several individuals that share an ideology and a similar set of beliefs. A few of politicians are said to be independent, i.e. with no political affiliation, though they can have affinity-sympathy with some parties. The members of a party are usually known by the society, as well as the ideology and the interests they support. Most politicians belong only to a single party, although some exceptions may exist, such as a politician that leaves one political party to adhere to another. For sake of simplicity, we assumed that a politician can only belong to one party during the period of analysis. Considering this condition, we define the following function:

Definition 4. *Be L a finite set of labels and $V^{'} \subseteq V$ a set of vertices from an interaction-triple graph whose labels are known. $\mathscr{L}$ is a mapping function $\mathscr{L} : V^{'} \longrightarrow L$ such that $\mathscr{L}(v) = l$ is the label for vertex v.*

This function returns as result a label (or political affiliation) for a given vertices (Person) in the interaction graph when the label is known. We assume that known labels are provided by an external source (we will explain this in section 4.2). The main task of party prediction is to complete the remaining unknown labels in the interaction graph. For such purpose, we took advantage of a label propagation process to expand the known labels to unlabelled vertices using only the graph structure. For example, given an unlabelled vertex v_i whose majority of neighbours are labelled as l_k, then v_i should also be labelled as l_k. Following this idea, we developed a propagation algorithm to diffuse labels across the interaction graph based on the following simple principle: *given a vertext v_i such that l_k is the label for the the largest proportion of its neighbours and l_k is greater than some random threshold, then v_i should also be labelled as l_k.*

In order to identify the neighbours for a vertex v who's label is l, we define the following function:

Definition 5. *Let be $\mathscr{N}$ a multi-valued function $\mathscr{N} : V \times \mathcal{G} \times L \longrightarrow V$ such that $\mathscr{N}(v, G, l) = \{w \in V_G : \exists (v, w) \in E_G \wedge \mathscr{L}(w) = l\}$ where $G \in \mathcal{G}$ and $\mathcal{G}$ is a set of interaction-triple graphs.*

Algorithm 1: Label propagation

```
1  Prograpagation (G, L, max)
   Input  : Graph (G), Set of label (L), maximum number of iterations (max)
   Output: Graph labelled (G)
2  for iter = 0, iter < max, iter++ do
3      G_c = G
4      for each vertex v in G do
5          Θ_v = random(0,1)
6          maxNeighbours = 0
7          label = -1
8          for each label l in L do
9              current = |neighbours(v, G, l)|/|allNighbours(v, G)|
10             if current > Θ_v then
11                 if current > maxNeighbours then
12                     maxNeighbours = current
13                     label = l
14                 end
15             end
16         end
17         if maxNeighbours > 0 then
18             G_c[v] =label
19         end
20     end
21     G = G_c
22 end
```

Now, for each vertex v and for each label l, we retrieve the whole vicinity of v whose label is l. If the proportion of nodes is greater than a random threshold, then the node is labelled accordingly. The Algorithm 1 describes the procedure employed in order to assign labels to unlabelled vertices. The algorithm input is composed by the graph of interest and the labels involved. We assumed a *neighbours* function(Definition 5) for identifying neighbours with the same label. The output is a copy of the original graph where vertices are labelled.

4 Implementation and Results

4.1 Data Collection and Preprocessing

The input data required to create an interaction-triple graph is a corpus of news articles. We collected such a corpus using a web crawling system. A web crawler is a piece of software able to identify hyper-links in web documents, retrieve such documents and store them properly. This process is repeated systematically until no new documents are found or a stop condition is satisfied. Such procedure enables us to explore the web by creating a graph-based representation of it.

Table 1: Corpus collected using a Crawling System. All dates correspond to 2016

Name	Seeds	Date	No. docs
Corpus 1	www.irishtimes.com/election-2016 www.irishtimes.com/news/politics www.irishtimes.com/business www.irishtimes.com/opinion	04 May	470
Corpus 2	www.irishtimes.com/election-2016 www.irishtimes.com/news/politics	11 May	65

The input for a crawling process is a small set of seed web pages. From these seeds, it is possible to cover a large portion of web documents. We selected seeds according to the news article we are looking for: politics and economics. We decided to restrict the retrieval process to a select set of domains in order to create an ad-hoc filtering. Table 1 presents seed web pages and additional statistical information for each crawling process.

There are several tools in the industry for performing crawling. We decided to use Nutch[2]. Nutch is a well-matured Apache project that has been developed over the past 10 years. The Nutch processing model has the form of a pipeline, in which for each input URL a series of steps are performed (fetching, parsing, storing, etc.). The original pipeline included by default in Nutch can be extended by connecting extra processing units or *plugins* to it. For our collection, we used plugins for text parsing, text indexing and data storing. This ability of extending the functionalities of Nutch was the main reason for our election.

The crawling process was focused only in the news portal *Irish Times*. We extracted news related only to politics and economics since there is a higher probability to obtain mentions of politicians in such news. We employed the following pipeline in Nutch: *i*) An initial URL (seed) is taken from the main queue. *ii*) The crawler fetches the URL, creating a binary representation of it. *iii*) Hyper-links are extracted from the web page and included in the main queue for later processing. *iv*) The web page is stored as a binary file in the storage backend. *v*) The crawler parses the content of the web page, obtaining a text representation of it. *vi*) The text representation, as well as other metadata fields, are stored as separated fields in the storing backend. This process is repeated until there is no page to retrieve or a stop condition is met. One requirement of Nutch is a storage system backend. Among the many alternatives available for Nutch, we selected Mongo[3] for its flexibility, its capacity to extend documents and the ability to perform text queries on the articles as well as map-reduce operations.

The flexibility of Nutch gave us the opportunity to try different parsers during the step *v*). Nutch includes several parser plugins. We selected two of them: *a*)

[2] http://nutch.apache.org
[3] https://www.mongodb.com

Standard Tika parser[4] and *b*) Boilerpipe parser [3]. The former did not produce suitable results since its output includes several text fields that are not related to the main story of the article, e.g. advertisement, link titles or section names. Such additional piece of text added noise to the article, increasing the complexity of NER. On the other side, Boilerpipe was developed for identifying the main section of a web page and extracting only text related to such section. This parser can identify the main story in a document and store only such a text in the backend. Since one of our assumptions is that a news article has one main story, we decided to use only this parser.

Once the web documents are collected and their texts are extracted, we use them as the input for a NER system. We implemented a NER system based on the well known GATE NLP Framework [2]. GATE allows us to define pipelines for information extraction (IE) that can be built using combinations of processing resources (PR) such as tokenisers, named entities extractors, POS taggers, language detectors and many more. In order to increase the interoperability of our NER system, we created a REST service for embedding our NER system. This REST service accepts as input an array of texts and returns an array of found named entities per each text in the input. Each entity returned contains the named entity text, the position where such entity is occurring in the text, its type class and additional meta-data fields. As we explained in section 3.2, we use named entity type property for filtering them out. Only Named Entities whose type is Person are kept for the next steps.

We analyse the selected named entities in order to identify duplicates, noise or irrelevant entities. For example, if the name of the entity consists of only one word, it is very hard to disambiguate it since several other entities can be similar to it. For this reason, we defined a series of rules to keep only entities that are simple to disambiguate. Once irrelevant entities are removed, we look out for duplicates in the entities. For this step, we created clusters of entities according to the Monge-Elkan similarity [1], in the following way:

i We group entities from the whole corpus into bins such that all entities within a bin have exactly the same name.
ii We iterate the bins one-by-one against a list of clusters. Initially the list is empty.
iii When the first bin is checked, a new cluster is created and the bin is added to that cluster.
iv For the following bin, we compute Monge-Elkan similarity between it and all current clusters in the list.
v If the highest similarity is greater than a threshold, the bin is assigned to that cluster. Otherwise, a new cluster is created containing only this bin.
vi This process is repeated for each bin. At the end, a list of clusters is returned, where each cluster is made up of a set of bins.
vii Lastly, for each cluster one of the bins is selected randomly and used as the cluster centroid.

[4] https://tika.apache.org

Table 2: Statistics for each graph.

Property	Corpus 1	Corpus 2
Vertices	1138	249
Edges	3725	1840
Connected Component (CC)	172	4
Largest CC (LCC) Vertices	534	239
Largest CC Edges	2513	1826
LCC Average Path Length	6.774	2.883
LCC Diameter	18	6
LCC Clustering Coefficient	0.762	0.632

Once the list of named entities is cleaned, we are in condition of creating an interaction-triple graph according to definition 2, in which each node represents a named entity and an edge between two nodes indicates that they are co-occuring in a document. This graph represents the interactions of named entities representing people that were found in the corpus.

The Table 2 shows some relevant graph statistics for each corpus. For computing such statistics, we used the Python library Graph-Tool[5]. In both cases, the interaction graph is similar to a Small World Network (SWN) [4]. SWNs are characterised by a low average path and a relatively high clustering coefficient. In a SWN, there are some vertices that act as short-cuts for vertices that are too far for each other. Figure 1 shows the trade-off between different random SWNs with the same size than the graphs created. Both interaction graphs are in the limit of what we can call a SWN. This is clearly seen in the degree distribution figure. The distribution curve is in between a Binomial and Power-law distribution.

4.2 Ground Truth Dataset

Party prediction requires a ground truth dataset. This dataset contains a list of people with their corresponding political affiliation or labels. We manually created this dataset using the following approach:

1. First, we selected a fixed number of most representative political parties for Ireland. We considered the number of members in the *Oireachtas* (The Irish National Parliament) as the criteria for determining the relevance of the party. The Irish Parliament is composed by 158 representatives (or deputies) and 60 Senators. Representatives compose the lower house of the parliament and they are elected by popular vote. Senators, instead, are elected by different political institutions in Ireland. We manually counted the number of current representatives for each party for assigning a relevance score. In this way, we selected the following parties: *Fine Gael*, 50 deputies; *Fianna Fin*, 43 deputies; and *Sinn Fin*, 23 deputies.

[5] https://graph-tool.skewed.de

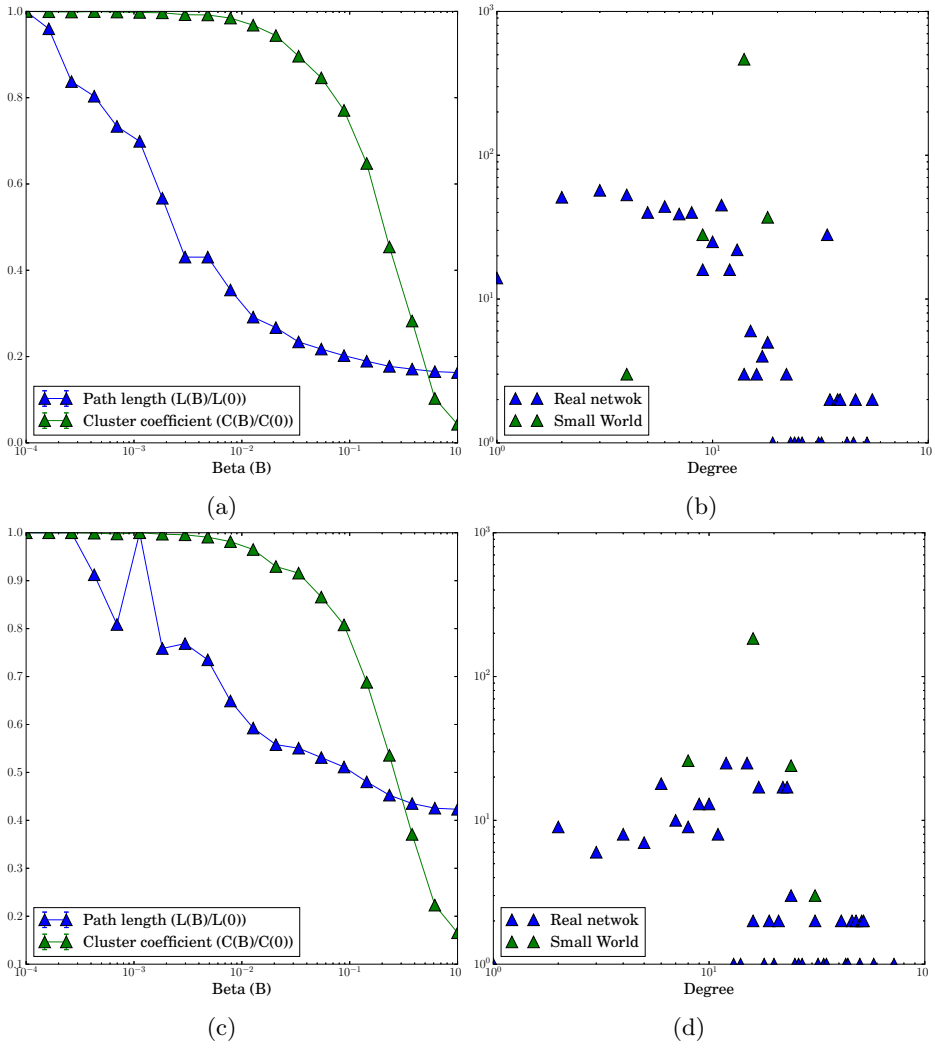

Fig. 1: Comparison of random SWNs and interaction-triple graph. Sub-figure (a) shows the variations on path length and cluster coefficient for small changes on parameter β for random SWN with similar size to interaction graph for corpus 1. Sub-figure (b) indicates the distribution for the actual interaction graph and a similar size random SWN. Sub-figure (c) and (d) show similar figure to (a) and (b) respectively but for corpus 2.

2. The names of the deputies were manually obtained and labelled. Once we obtained the names, we used them for determining a score for each deputy, so we can rank them accordingly. The score is proportional to the frequency

of mentions for a given deputy across one corpus, i.e. deputies that are quiet frequent in news articles are ranked higher.
3. We obtained the top-k deputies for each label from the ranking. We randomly selected h deputies for each top-k list with the only constriction that $h << k$. This constriction is necesary in order to over label our graph. The combination of h deputies is the input for our label propagation algorithm. We selected $k = 10$ and $h = 5$.
4. Finally, we used Elkan-Monge similarity to identify which vertices in the graph are equivalent to those elements in input list.

For asserting a positive match, we linked each name in ground truth to a vertex in the graph such that the similarity is maximum. We computed a weighted average precision and recall. We ran the algorithm several times in order to see its tendency using different inputs. We use 0.9 as the threshold for Elkan-Monge similarity.

4.3 Results

Once the interaction graph and the labels are produced, we use the Algorithm 1 in order to predict labels. We iterated this algorithm until most vertices were labelled. We executed our algorithm several times and we computed the microaverage for each metric (Table 3 shows the average results obtained). The results were worse in corpus 1 than corpus 2. We think that this is due mainly to the presence of noise in corpus 1. In corpus 1, we have a higher probability to generate noise than corpus 2 since the crawling process is open to more noisy news articles. We used different threshold for Elkan-Mong similarity comparison, and we obtained similar results. Other factors that may contribute to the performance are the temporality of the corpora. For instance, it is also possible that during the time of the collection of corpus 1 the number of politicians referred on the ground truth was less than the period of corpus 2.

In general our results reveal that the interaction graph has some degree of relations according to the party structure. But, it seems that several other features may affect the mentions of politicians in news articles, in particular the filtering applied during the collection of news directly impacted the results. However, other factors such as locality, temporality, support for same law projects, can also play an important role in the results. We believe that an increment of the volume of the news articles may improve the precision and recall as well as a more refined filtering, since a bigger interaction-triple graph can reflect more clearly some properties of co-occurrences.

Besides, the ground truth set designed for our experiments is inadequate for capturing the structure of political interactions. Our approach is restricted to the current members of the Irish Parliament and does not cover politicians that do not hold a position in the Parliament. An extension of the ground truth can reveal the power of prediction model by considering more people in the evaluation.

Table 3: Precision (Pre), Recall (Rec) and F1-score (F1) for Party prediction using our Label propagation algorithm.

Property	Pre	Rec	F1
Corpus 1	0.398	0.350	0.316
Corpus 2	0.454	0.469	0.414

5 Conclusions

We presented a graph model for mentions of named entities in news articles. Our model uses the co-occurrence of named entities in order to create such representation, and selects only named entities whose type is person. With this graph, we proposed a propagation algorithm for predicting the political affiliation of named entities, considering that we crafted a corpus related mainly to politics. Our results revealed that there is a correlation between the political affiliations of people and their co-occurrence in news articles. We think such correlation makes sense since most news articles reflect the political debates and interactions, and these debates are centred around the government-party structure. Our evaluation considered two different corpora created from the same news portal. One corpus was larger than the other, but also contained more noise. Our findings reveal that the best prediction results are obtained with that corpus much more refined.

There are several tasks to be followed in the future. For instance, it is of interest to know how the number of news articles affects the power of the prediction model as well as the size of text chunk. Our evaluation compared only to one government structure (the Irish Parliament), however it will be interesting to integrate additional structures in our evaluation. Besides, our model can be the input for identifying structures in the political interactions that are not trivial or intuitive. Finally, it will be of major interest to try different interaction types rather than only mentions (co-occurrence).

References

1. Cohen, W., et al.: A comparison of string metrics for matching names and records. In: Kdd workshop on data cleaning and object consolidation. vol. 3, pp. 73–78 (2003)
2. Cunningham, H.: Gate, a general architecture for text engineering. Computers and the Humanities 36(2), 223–254 (2002)
3. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 441–450. ACM (2010)
4. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-worldnetworks. nature 393(6684), 440–442 (1998)