Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A system for extracting and comparing memes in online forums

Héctor Beck-Fernandez[a], David F. Nettleton[b,*], Lorena Recalde[b], Diego Saez-Trumper[b], Alexis Barahona-Peñaranda[a]

[a] Department of Computer Engineering, Universidad de Tarapacá, Arica, 1010069, Chile
[b] Department of Information Technology and Communications, Universitat Pompeu Fabra, c/Roc Boronat, 138, 08018 Barcelona, Spain

ARTICLE INFO

Article history:
Received 14 November 2016
Revised 3 April 2017
Accepted 4 April 2017
Available online 5 April 2017

Keywords:
Text processing
Semantic networks
Online forums
Information retrieval
Search techniques
Information and knowledge
Computational Intelligence

ABSTRACT

From their origins in the sociological field, memes have recently become of interest in the context of 'viral' transmission of basic information units (memes) in online social networks. However, much work still needs to be done in terms of metrics and practical data processing issues. In this paper we define a theoretical basis and processing system for extracting and matching memes from free format text. The system facilitates the work of a text analyst in extracting this type of data structures from online text corpuses and n performing empirical experiments in a controlled manner. The general aspects related to the solution are the automatic processing of unstructured text without need for preprocessing (such as labelling and tagging), identification of co-occurences of concepts and corresponding relations, construction of semantic networks and selecting the top memes. The system integrates these processes which are generally separate in other state of the art systems. The proposed system is important because unstructured online text content is growing at a greater rate than other content (e.g. semi-structured, structured) and integrated and automated systems for knowledge extraction from this content will be increasingly important in the future. To illustrate the method and metrics we process several real online discussion forums, extracting the principal concepts and relations, building the memes and then identifying the key memes for each document corpus using a sophisticated matching process. The results show that our method can automatically extract coherent key knowledge from free text, which is corroborated by benchmarking with a set of other text analysis approaches, as well as a user study evaluation.

1. Introduction

Nowadays, most online newspapers offer online comments forums for published articles. For example, on 8th March 2016 an article was published in the Washington Post (http://www.washingtonpost.com/) which dealt with the US presidential primary elections. In a period of 3 h, the comments forum had already accumulated 900 individual posts. However, it is unlikely that any one individual is going to read the 900 (or many more) posts to evaluate the reader's opinions about the theme posed by the article. But we propose that it would be very useful to know which are the most important ideas represented in the comment threads. By looking at the latest page of comments, we can see that some of the ideas include "Trump has a weaker hold on his party", "the GOP will lose the election to Hillary" and "Clinton has been widening her lead (on Sanders) in individual states such as ….." . We can also see if the author's ideas and those of the comments are shared. However, we cannot know all the ideas given by all the 900 posts nor manually synthesize (in a reasonable time) which are the key informational structures in the whole corpus of posts.

What are the recurring ideas which propagate through the comment threads? It would be very useful to know the answer to this, and other related questions. In this paper, we postulate that the main ideas (concepts and their relationships) of free text forums can be represented as minimal semantic networks which are transmitted through the comments as 'memes', where a meme is defined as a unit of transmittable knowledge. We consider a given semantic network as 'minimal' when inasmuch as possible it does not have repetitions of concepts or relations and has a minimum diameter in a graph structure sense (this will be defined in Section 3).

Memes are a relatively new form of representation for the transmission of knowledge between individuals, which offers a promising approach to better understanding and defining the way in which ideas evolve. However, much work still needs to be done in terms of their definition, representation and data processing for real scenarios.

[*] Corresponding author.
E-mail addresses: hbeck@uta.cl (H. Beck-Fernandez), david.nettleton@upf.edu (D.F. Nettleton), lorena.recalde@upf.edu (L. Recalde), diego.saez@upf.edu (D. Saez-Trumper), ale.genbp@gmail.com (A. Barahona-Peñaranda).

http://dx.doi.org/10.1016/j.eswa.2017.04.010

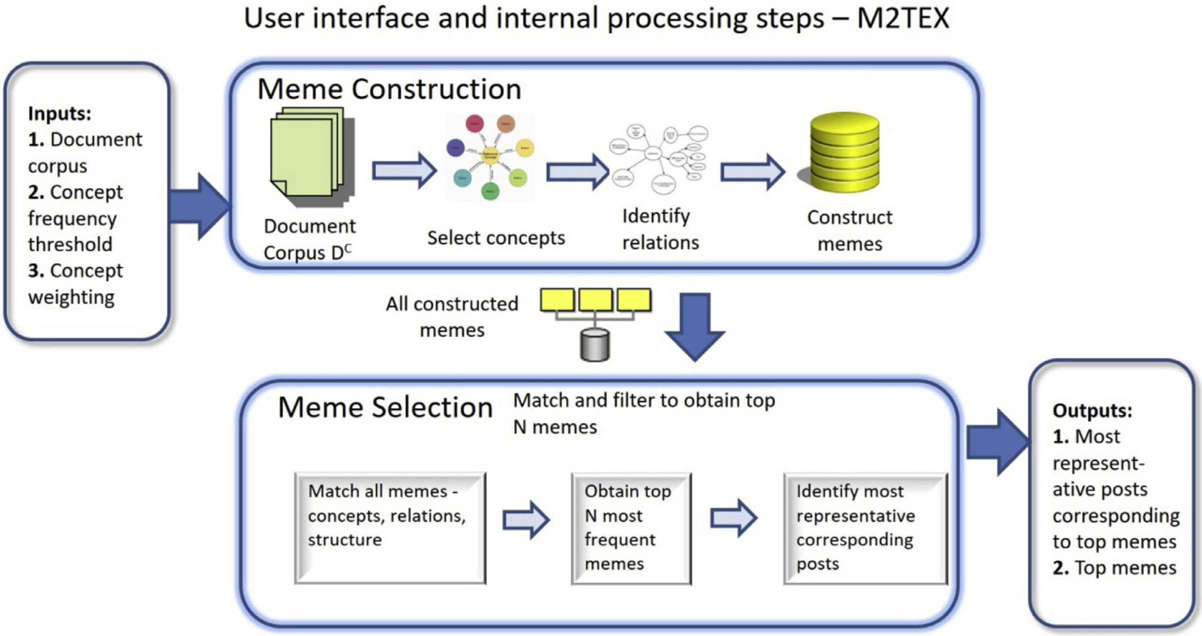
User interface and internal processing steps – M2TEX
Inputs:
1. Document corpus
2. Concept frequency threshold
3. Concept weighting
Meme Construction
Document Corpus D^C
Select concepts
Identify relations
Construct memes
All constructed memes
Meme Selection
Match and filter to obtain top N memes
Match all memes - concepts, relations, structure
Obtain top N most frequent memes
Identify most representative corresponding posts
Outputs:
1. Most representative posts corresponding to top memes
2. Top memes

Fig. 1. Schematic diagram of the system.

The present research is motivated by the scarcity of work in the field of extracting full semantic networks (that is, concepts and corresponding relations) from raw free format text, and their post-processing in order to identify key information structures. Also, a new field of study is offered by the matching and interpretation of minimal semantic networks extracted as memes.

The main idea of the proposed process is to build semantic network type structures from free text, finding the essential ones from many such structures which are identified. Then we find “minimal structures" from the semantic networks which have the highest occurrences (as memes) in the posts of the text corpus. The result and the objective is to obtain a subset of “memes" which represent the core concepts being discussed in the discussion posts. This enables the user to obtain a summary without having to read the often hundreds of posts associated with a news article.

Hence, in the present work we have developed a system which proposes to cover these issues: **(i)** the automatic identification and extraction of key concepts and relations, **(ii)** the construction of semantic networks (memes) from the concepts and relations, and then choosing the top subset of memes for a given document corpus, and **(iii)** facilitating the meme extraction process for a user who is not a text analysis specialist, or for expert users to reduce time dedicated to text processing, calibration and experimentation. Making this system available to the text analysis community will have a big impact in impulsing online text corpus analysis.

The objective of the present work is first to algorithmically define a set of processes which will facilitate the identification and extraction of key knowledge structures in the form of semantic networks from free text. This represents a challenge in terms of identifying contextual meaning in the presence of unstructured, free-format text which contains noise and errors. Secondly we filter the extracted semantic networks to identify the key minimal sets which represent the recurrent memes. This second step embodies a novel meme matching process. We take ideas from two fields, those of semantic networks and information retrieval, in order to identify the key memes from a candidate set, which are most relevant and have the best coverage with respect to a “relevant set”. Real examples of online discussion forums of comment posts are processed to illustrate how the framework can be applied in practice.

Fig. 1 shows a schematic representation of the user interface and internal processing modules. Firstly, in the “meme construction” module, key concepts and relations are extracted from the document corpus, using a frequency based filtering to identify the most relevant. The concepts and relations are then used to construct a population of memes. In a second “meme selection” module, the memes are matched in order to find groups of memes which are similar (that is, derived from the same “root meme”). Next, the top N memes are selected based on frequency of the “root memes” in the meme population. Finally, the most representative texts are summarized from the corpus which correspond to the top memes. The user interface is simplified for non-expert users and requires three parameters: the document corpus to be processed, a concept frequency threshold (low, medium, high) which decides how many concepts to include based on their frequency and a concept weighting (low, medium, high) which decides the relative importance of concepts to relations. User tips and help can be supplied to explain to a novice user how to assign these values, and default assignments are given. For expert users, an “advanced options" mode allows access to all control parameters of the system, which are explained in Sections 3 and 4 of the paper. In contrast with other methods (in which after each step manual intervention is required), our system processes the text in a continuous automated sequence of steps and also does not require preprocessing of the input text (i.e. tagging or labelling) by the user.

The primary contributions of the paper are:

- System which allows a non-expert user (that is a text analyst not specialized in semantic meaning extraction) and/or an expert user (facilitating his/her work) to extract the key memes which are most representative of an article and its corresponding online discussion forum. With reference to Fig. 1, this can be done by varying the two main control parameters, the concept frequency threshold α and the relative weighting of concepts to relations ρ. From this the trade-off between a wider inclusion of topics (frequency threshold) and the importance of concepts versus relations (weightings) can be evaluated.

- Other parameters are pre-assigned with values that have been found to be optimum for different document corpuses, although the expert user can modify them in advanced mode. These parameters are: n-gram size *NS*, context window size *CW* for extracting concepts and relations and similarity thresholds for matching $\{\varepsilon, \theta, \tau\}$.
- Automatic extraction of concepts and relations and building of a semantic network (memes) population for a given document corpus (online discussion forum).
- Selection of top N subset of memes which are most representative of key discussion themes in a document corpus. Selection of corresponding posts to the top memes which are shown to the user as the core discussion issues.

Secondary or auxiliary contributions of the paper are:

- An integrated sophisticated semantic network extractor which given a document corpus, extracts the key semantic structures which are most representative of an online article and its associated discussion forum.
- A new approximation to the problem of automatic extraction and matching of semantic networks consisting of the most highly relevant and interrelated concepts and relations.
- A comparison of our approach with a set of metrics well known in the text analysis field.
- Efficient matching algorithm to identify similar semantic network groups each represented by a prototype (“root meme”), and then to select the top N prototypes.

The structure of the paper is as follows: in Section 2 we present the state of the art and related work; in Section 3 we present the theoretical definitions for the meme environment (documents, concepts, relations between concepts, and memes), their identification in a text corpus and their comparison; in Section 4 we present the results of data processing using our method on four extensive discussion forums; in Section 5 we benchmark different metrics and approaches from the state of the art on the same document corpuses; finally, in Section 6 we summarize the current work and present some conclusions.

2. Related work

In the following section we will first briefly comment some examples of relevant expert systems, followed by a more detailed review of related research in the field of knowledge extraction from text corpuses, which is the principal focus of our current work; the section is then finalized by discussing the relation of our proposed method to those of the state of the art. Throughout the following sections, we contrast existing approaches with the novel aspects and characteristics of our system.

2.1. Relevant expert systems

In this subsection we will briefly comment selected exemplary expert systems whose design and user objectives can be extrapolated to the current work.

In Mooney and Nahm (2003) a text mining system is presented which searches for patterns in unstructured text. A framework for text mining, called DISCOTEX (Discovery from Text EXtraction), uses a learned parameterization to preprocess text into a more structured form, which is then analyzed to identify “interesting” relationships. The DISCOTEX system is comprised of an IE (Information Extraction) module together with a rule induction module. Also, rules derived from a database built from different text corpus are used to predict additional search information to extract from future documents, thus improving the recall of the underlying extraction system. The system is applied to a corpus of computer job announcement postings from an Internet newsgroup, and results gave over 90% precision and over 80% recall on average. This was achieved by training first on a labeled set of postings and then testing on unlabeled examples. However, we might say that the postings represent a semi-structured information representation rather than free text, and the simply structured rules extracted were comprised of named entities (nouns) with a well defined taxonomy.

Peng, Gao, Zhu, Huang, Yuan, and Li (2016) present an information extraction method and query-oriented summarization for automatic query-reply in social networks. The authors state that a key consideration is the filtering of redundancy and noise in the raw data, which are two of the causes of poor reply quality for social network messages. The system is comprised of two main modules: (i) an information extraction module based on time-frequency transformation; (ii) a query-oriented text summarization module which produces brief and concise summaries as the final reply. This is based on the scoring, ranking and selection of the sentences extracted by the first module. We observe that the social network text messages are taken from the Sina Weibo app, for a preselected set of topics, where the messages are short texts (up to 140 words). A feature matrix with frequency scores is used, which is similar to our concept matrix, and similarity function is defined to calculate the distance between a given sentence and a query. Sentences are scored using three subscores: (i) content weight, (ii) location weight and (iii) similarity between sentence and query. Each of these subscores is weighted using three respective weights, alpha (α), beta (β) and gamma (γ), to balance the overall score and sum the subscores. Their system is characterized by the following parameter settings: top message set size $T = 1000$; top selected sentence set size $m = 5$; sentence length threshold sigma $(\sigma) = 5$; subscore weightings $\alpha = 0.5$, $\beta = 0.3$ and $\gamma = 0.2$ respectively; similarity threshold for sentence selection tau $(\tau) = 0.7$. The results were evaluated by precision, recall and F-score measures with respective average values of 0.66, 0.45 and 0.53 over two different topics.

In Thangaraj and Sujatha (2014) an architectural design is presented for effective information retrieval in the semantic web. Their premise is that information retrieval based on keywords is insufficient in order to retrieve documents from the semantic web, and their solution is to incorporate the “concept” to enrich the search information. They adapt the classic TF/IDF metrics to include the concept with the keywords, and the concepts are elicited using the OWL ontology thesaurus. Thus a weighted score is obtained between a given query Q and a document D_i. The architecture of the system is designed with five layers: application, retrieval, semantic matching, semantic annotation and physical data. Two of the key original algorithms described by the authors are the “semantic annotator and indexer” (semantic annotation layer) and the “semantic query conversion and ranked retrieval” (retrieval layer). The average precision and recall for the semantic based retrieval was 0.989 and 0.685, respectively, whereas for the keyword based retrieval the precision and recall were 0.451 and 0.552, respectively. We note that these tests used a manually prepared set of queries. The test data were web-documents obtained by the crawler on pre-selected domains and downloaded onto the local computer. In terms of user definable parameters, the *t* value defines the threshold for including results and the concept ontology.

Rubio, De la Sen, Longstaff, and Fletcher (2013) present a modular expert rule based system which facilitates the automatic selection of cutting parameters in milling operations. Although this represents a different data modeling domain problem to our own, their system is exemplary in terms of modular design, parameter calibration and end user motivations, which can be generalized to the current work. The system defines a cost function in terms of five key control parameters, which are associated with the milling process setup. In our case the control parameters are the data definitions and the desired privacy level, whereas the cost function measures the information loss due to anonymization.

2.2. Meme processing

The term “meme" was originally defined by Dawkins (1989), and has been recently applied to the study of how information spreads through Internet and Online Social Networks (OSNs).

According to Dawkins (1989), a “meme" or a “memetype" is similar to a “gene" or “virus". It consists of a basic unit of information circulating within a community, and research from a social sciences perspective has studied how it serves as a mechanism to propitiate cultural and social evolution. Heylighen and Chielens (2013) compare the ‘meme’ with the ’gene’ and formalize the following meme properties: ‘longevity’, the duration that an individual meme survives; ‘fecundity’, the reproductive activity of a meme; ‘copy-fidelity’, the degree to which a meme is accurately reproduced.

Bordogna and Pasi (2013) proposed a schematic definition for memes using an OWL schema, followed by the definition of several operators to extract memes from online blog posts using information retrieval methods and n-grams (contiguous sequences of n items from a given sequence of text). A fuzzy-type matching is performed to evaluate the fidelity of a given blog post to an original meme description. Finally, the longevity is considered by ordering the text entries by their timestamp and taking into consideration the fidelity. The system used the Google search engine (blog search option) to find posts potentially related to pre-defined ‘ememe’ phrases.

Leskovec, Backstrom, and Kleinberg (2009) developed a framework for tracking short textual memes in an online news media environment, identifying a general class of memes. Simmons, Adamic, and Adar (2011) presented a study about meme mutation in social networks. They uncovered patterns in the rate of appearance of new variants, their length and popularity, and developed a simple model that is able to represents these attributes. Nettleton (2013) presents a wide-ranging survey of OSN analysis, covering themes such as ‘influence and recommendation’ and ‘information diffusion’, which includes contextual entity tracking using memes. Baydin and López de Mántaras (2012) present an evolutionary algorithm based on the concept of memes. They used semantic networks to represent the individual pieces of information, and employed the ’genetic’ concepts of crossover and mutation to model changes over time.

2.3. Semantic knowledge extraction from text corpuses

Different approaches exist for extracting semantic knowledge from text corpuses. Szumlanski and Gomez (2010) extracted semantic networks based on frequency and concept affinity, from Wikipedia texts using the WordNet (2010) ontology database to identify related concepts. Jiang and Conrath (1997), on the other hand, describe a semantic similarity metric based on corpus statistics and a lexical taxonomy. They present an approach for measuring semantic similarity/distance between words and concepts which uses a distributional analysis of the corpus data. Chen, Gangopadhyay, Karabatis, McGuire, and Welty (2007) deal with the elicitation of semantic networks based on concepts relevant to the data mining of specific datasets. Kok and Domingos (2008) present an unsupervised approach to extracting semantic networks from large volumes of text. They used the TextRunner system, as described in Banko, Cafarella, Soderland, Broadhead, and Etziono (2007), to extract tuples from text, and then induce general concepts and relations from them by jointly clustering the objects and relational strings in the tuples. Their approach is defined in Markov logic using four basic rules to extract meaningful semantic networks.

Two systems which represent contrasting approaches for extracting key knowledge from free text document corpuses are Pingar (Medelyan & Divoli, 2012) and Unitex (Muniz, Nunes, & Laporte, 2005). On the one hand, Pingar represents a robust commercial system which has a utility for text summarization and keyword/key-phrase identification. Unitex, on the other hand represents a toolkit for text mining which has been used widely in the academic field. Medelyan and Divoli (2012), present the Pingar system (www.pingar.com), a system for mining unstructured textual data. Pingar is based on algorithms which use machine learning, text processing and mining, statistics, NLP and linguistic methods to perform text analytics. One singular function of Pingar is the summarization feature which can be used in the online tool demo by giving plain text as input. Pingar will detect keywords and key-phrases from the text and then uses them in context to extract a summary. The summary is composed of a small number of few paragraphs that are chosen depending on the weight of the identified keywords. Unitex (Muniz et al., 2005), is widely used for Natural Language Processing in University research. It provides many functions which can be programmed to perform text mining and linguistic analysis. According to the authors, the library of language resources Unitex, provides the grammar dynamism needed by a language. The flexibility added by some lexical functionalities makes it possible to put a word into context and analyze the concepts around the word taking into account the number of cooccurrences and z-score. In the current work we have used Unitex for benchmarking comparisons using the frequency and z-score cooccurrence metrics. Given that Pingar and Unitex (version 3.1 beta) represent two different approaches for state of the art text mining, we chose them to perform the benchmarking comparisons which are described later in Section 5 of the paper.

The definitions and work in the current paper are evolved from the initial more elemental schemes, which the authors presented in Beck-Fernandez and Nettleton (2013). We observe that many systems do not process automatically and require considerable manual intervention in between the different processing phases, whereas our method executes its three comprising steps in sequence, without intervention, the output of each step serving as input to the next.

2.4. Relation of the proposed method to those of the state of the art

In this subsection we clarify the relation of the method we propose to the ones mentioned in the state of the art. A meme can be considered a special type of semantic network, in which overall frequency and compactness of the resulting semantic network are key considerations (Baydin & López de Mántaras, 2012; Leskovec et al., 2009; Simmons et al., 2011). In contrast, the general extraction of semantic networks does not emphasize these aspects (Chen et al., 2007; Kallipolitis, Karpis, & Karali, 2012; Kok & Domingos, 2008; Oh, Kim, Park, Yu, & Lee, 2013). To the best of our knowledge there is a scarcity of scientific work dedicated to identifying and extracting memes in free-text online content. Some systems try to extract semantic networks with a focus on named entity relations, whereas our focus is on the identification of concepts in a more general sense. In the case of Bordogna & Pasi (2013), real blog posts were processed, however, Google search was used to find posts which matched a pre-defined SQL type query made up of keywords, and the matching was essentially keyword based. In contrast, our system uses an integral post forum related to a specific news article, without any pre-filtering.

Other systems do not process real data and use synthetically generated examples (Baydin & López de Mántaras, 2012) or rely on well written semi-structured text corpus such as Wikipedia

Table 1
Summary of advantages and inconveniences of systems and methods in state of the art.

Author reference	Advantages	Drawbacks
Baydin and López de Mántaras (2012)	Builds meaningful semantic networks	Only tested on synthetically generated examples
Szumlanski and Gomez (2010), Oh et al. (2013), Jean-Mary et al. (2009)	Metrics can be adapted for non-structured text	Rely on well written semi structured text corpus such as Wikipedia or Wordnet
Leskovec et al. (2009), Simmons et al. (2011), Heylighen and Chielens (2013)	Process online news articles. Leskovec et al. and Heylighen & Cheilens consider more detailed meme metrics	Only process article, not forum posts
Pingar system: (Medelyan & Divoli, 2012)	Easy to interpret for non- expert end user	Only named entities identified
Unitex system: (Muniz et al., 2005)	Effective for applying specific metrics and obtaining specific results (e.g. co-occurrences with z-scores), collocations.	- Requires expert user text analyst -Only co-occurrences, no function to identify relations - No thresholds as cut-off for frequency/z-score
Bordogna and Pasi (2013)	Processed real forum posts	Google search used to find matches to SQL queries - does not have own matcher
M2TEX system	- Processes real online news articles and related post forum without any pre-processing. - Consecutive steps are automated	User may only want part of complete processing (e.g. ranked list of concept pairs) - Does not quantify detailed meme metrics

(Szumlanski & Gomez, 2010), PubMed (Oh et al., 2013) or Wordnet (Jean-Mary, Shironoshita, & Kabuka, 2009). Other systems have analyzed online news media (Leskovec et al., 2009), but have focused on edited content (that is the newspaper articles) whereas we also process the comments forums. In contrast, our system analyzes real user posts in forums with no previous formatting or processing. Other systems only perform entity extraction or require programming and/or human interpretation in order to identify co-occurrences (Medelyan & Divoli, 2012; Muniz et al., 2005). Also, many systems do not process automatically and require considerable manual intervention in between the different processing phases. Our method executes four steps in sequence, without intervention, the output of each step serving as input to the next.

One limitation of the method we present is that it does not explicitly calculate the meme metrics of ‘longevity’, ‘fecundity’ and ‘copy-fidelity as the final step of the process (Heylighen & Chielens, 2013). However, these characteristics are implicitly considered when we perform the matching phase to choose the top memes.

Hence, there are no systems in the literature which follow a directly comparable method to our complete processing scheme. Thus, we benchmark the first processing phases (using the metrics from Unitex and the Pingar system commented previously) which extract the concepts, relations and co-occurrences which are essential for building the semantic networks and identifying the most frequent and compact memes.

To summarize, in Table 1 a comparison is show of the related works and our system M2TEX, including the advantages and drawbacks of each.

3. Theoretical framework of the meme extraction system

In this Section, we give the form to represent the memes, which allow us to describe, represent, extract and compare them in free text documents. We note that the form of representation of a meme throughout is as a semantic network, composed of tuples of concepts and relations between the concepts. Beck-Fernandez and Nettleton (2013) presented some initial definitions and algorithms. Section 3 is organized as follows: in Section 3.1 the basic definitions are given which are required for the processing framework; in Section 3.2 a description is given of how the text is processed to identify the concept and relation tuples; in Section 3.3 a description is given of how the concepts, relations and memes are matched and how memes are constructed; finally, in Section 3.4 definitions are given which are used to identify the top memes. Refer to Fig. 1 for a depiction of the overall processing scheme.

3.1. Basic definitions

In this subsection, we give the form to represent the memes, which enables us to identify them in a document corpus, based on an initial set of concepts and relations.

Let $D = \{d_1, \ldots, d_n\}$ be the set of all documents or posts written in a free text online forum; $C = \{c_1, \cdots, c_k\}$ is the set of concepts extracted from D; $M = \{m_{1,1}, \ldots, m_{i,j}\}$ is the set of all memes found from D, where m_{ij} is the jth meme of the ith document; $R = [r^{k,l}_{i,j}]$ is an array ordered by meme, where $r^{k,l}_{i,j}$ is the relationship between the concepts c_i and c_j contained by meme $m_{k,\ l}$, of all relationships extracted from D. Note that a relationship may be in more than one meme. We will use the sub-index “0” to indicate that a meme contains just one concept and therefore has no relationships. For the purposes of calculation, we assume the possibility of an ‘empty’ relationship, denoted by $\emptyset$. Each meme is a labelled directed graph comprised of edges (relationships) and vertices (concepts). Let e_{ij} and v_{ij} be the sets of edges (ordered pairs) and vertices of the meme m_{ij} respectively. In addition, M will be the set of all memes.

With reference to Fig. 1, the system has two external (user defined) and five internal control parameters: external - concept frequency threshold α, weight for concepts ρ; internal - n-gram size NS, context window size CW, concept similarity threshold ε, relation similarity threshold θ, meme similarity threshold τ. The assignment of these parameters is explained later in Section 4.1, Design of Experiments, and each is described in its context in the following definitions of Sections 3.2 and 3.3.

3.2. Concept and relation extraction

Let $IC(c_k)$ be the Information Content of the concept c_k, using the standard information theory definition, which is calculated as shown in Eq. (1):

$$IC(c_k) = -log_2[p(c_k)] \tag{1}$$

where $p(c_k)$ is the probability of encountering an instance of concept c_k, which may be calculated as shown in Eq. (2):

$$p(c_k) = \frac{freq(c_k)}{N} \tag{2}$$

being $freq(c_k)$ the number of occurrences of concept c_k and N being the total number of words observed. As detailed in Resnik (1999) and Jiang and Conrath (1997), and following the standard definition of information theory (Ross, 1976), the information content of a concept c can be quantified as minus its the log likelihood, −log p(c). Intuitively, as probability increases, informativeness decreases, hence more frequent concepts will in general, have a lower information content.

At this point we introduce the first threshold, designated as α, which indicates the required percentage of times that an n-gram must appear in the complete document set in order for it to be considered a concept, thus, $p(c_k) > \alpha$. This threshold has a default value which can be modified by the user.

Once the most frequent concept set has been identified for a given corpus, the next step is to find pairs of concepts which co-occur in the text and which belong to the most frequent concept set. A concept pair must be contiguous, that is, belong to one document and, furthermore, the concepts must be contained between full stops in the same sentence, paragraph or oration. This is because the concepts should be within a given context, that is, that two concepts 'are related or have a significance' within a paragraph, phrase, sentence or oration.

Algorithm 1 : Concept extraction.

```
Input: Set of documents $D \neq \emptyset$, set of stopwords $SW$, frequency
    threshold $\alpha \in (0, 1)$, n-gram size $NS$
Output: Set of concepts $C$, set of documents $D$
Process:
let $C \leftarrow \emptyset$; $D \leftarrow \emptyset$;
for each $w \in D$
  if $w \in SL$ then $D \leftarrow D - \{w\}$;
for each n-gram $w_n \in D$
  calculate $f(w_n)$ that represents the number of documents which
      contain the n-gram $w_n$;
if $\frac{f(w_n)}{|D|} > \alpha$ then $C \leftarrow C \cup \{w_n\}$;
```

Identification of relations: In the current paper it is not our focus to enter into details of text processing. However, during the implementation of Algorithms 1 and 2 it has been necessary to resolve problems associated with the identification of the most important relation between two concepts. In the following we give some details of how we have resolved this problem, with a schematic outline of the heuristic. This process corresponds to line 26 of Algorithm 2 (see Annex). The heuristic for the identification of a verb associated with two concepts is as follows:

- If there is only one unique verb, then select it.
- If there is more than one verb, then select the least frequent of the verbs.
- If there exist two consecutive concepts without verb, then select a verb which is contiguous to them
- (going rightwards), until CW words (CW=context window) have been counted or a full stop is reached.
- If there are no verbs which can be selected, then we have an empty relationship.

We note that previous to processing, stemming is applied to all verbs.

To compute the memes and the relationships we note that m_{ij} is the jth meme of the ith document and $R = [r^{i,j}_{k,l}]$ is an array ordered by meme; $r^{i,j}_{k,l}$ is the relationship between the concepts c_k and c_l of meme $m_{i,j}$, for all relationships extracted from D. The pseudo-code for the meme construction process is given in Algorithm 2 which is detailed in the Annex of this paper.

3.3. Memes - ontological and structural similarity

Meme construction and comparison is an essential component of the overall processing given that the frequency of memes has to be calculated in order to establish the most frequent and the most compact (or minimal).

In the following, we are going to consider two aspects of meme similarity: (i) ontological and (ii) structural. We propose that the ontological aspect has the primary weighting for meme similarity whereas the structure is a secondary aspect. For example, two memes may have the same structure, in terms of number of vertices and their connectivity, but the concepts are totally unrelated. Hence, we first look for a subset of memes whose concepts are as similar as possible, and from those we then choose which have the closest topological structure. The pseudo-code of Algorithm 3 is given in the Annex to this paper, which is an embodiment of the meme matching process described in this subsection.

3.3.1. Ontological similarity

Firstly, for our case, we will consider the definition of the semantic distance between two concepts as defined in Jiang and Conrath (1997), who proposed a new way to measure the similarity/distance semantics, which is based on a combination of an edge-based scheme initially proposed by Resnik (1999) with a vertex-based scheme proposed in Sussna (1993). If we consider the case when we only considered strong links, the distance is calculated as shown in Eq. (3):

$$dist\left(c_i, c_j\right) = IC(c_i) + IC\left(c_j\right) - 2 \times \; IC\left(LSup\left(c_i, c_j\right)\right) \tag{3}$$

where $IC(c_k)$ and $p(c_k)$ are as defined previously in Section 3.2 and (c_i, c_j) denotes the lower super-ordinate of c_i and c_j.

Different metrics exist for calculating the semantic distance, many of which are designed for specific ontologies, a review of which is given in Resnik's highly referenced paper (Resnik, 1999). In our case, we chose a metric which is well explained in Jiang and Conrath (1997), and which is adequate for processing free format unstructured text, a key characteristic of our current work. Jaing and Conrath's metric uses the WordNet lexical database which, due to its tree structure, facilitates the calculation of the "Information Content" for each of the concepts and relations which compose a semantic network.

In Fig. 2 we see a taxonomy based on WordNet of some example concepts extracted from a text. From this, for each basic concept (leaf of the taxonomy), we calculate its frequency, the probability of occurrence, and its Information Content (IC). The frequencies of concepts were computed using noun frequencies from The Corpus of Contemporary American English described by Davies (2010), subset 2010–2012, and the distances, were computed following the methodology of Jiang and Conrath (1997) and Resnik (1999). Hence, this procedure allows us to calculate the distance between concepts.

For a better interpretation of the values of the distances, we propose the concept of normalized distance, to obtain values between 0 and 1. It can be shown that the minimum value is 2.0 and the maximum is $log_2 N^2$, which gives the following Eq. (4):

$$dist_{con}\left(c_i, c_j\right) = \left(dist\left(c_i, c_j\right) - 2\right) / log_2 N^2 \tag{4}$$

where N is the Corpus size under consideration.

Definition 1. Similar concepts

We say that in a context, two concepts c_i and c_j are similar if $dist_{con}(c_i, c_j) < \varepsilon$, where $\varepsilon \in [0, 1]$ is the concept similarity threshold; otherwise the concept pair $\{c_i, c_j\}$ is semantically irrelevant. We note that c_i or c_j *may* be semantically relevant in the context of another concept pair, such as $\{c_i, c_k\}$ or $\{c_j, c_l\}$.

Relationships: In the present work, we need to measure the similarity between verbs in text, which are relationships between two concepts in a semantic network, and which represent key ideas within a document. For this purpose, we will use WordNet to calculate the distance between two verbs. This allows us to give the following Definition 2.

Definition 2. Similar relationships.

We say that the relationships r_i and $r_j$are similar if $dist_{rel}(r_i, r_j) < \theta$, where $\theta \in [0, 1]$ is the relation similarity threshold and r_i and

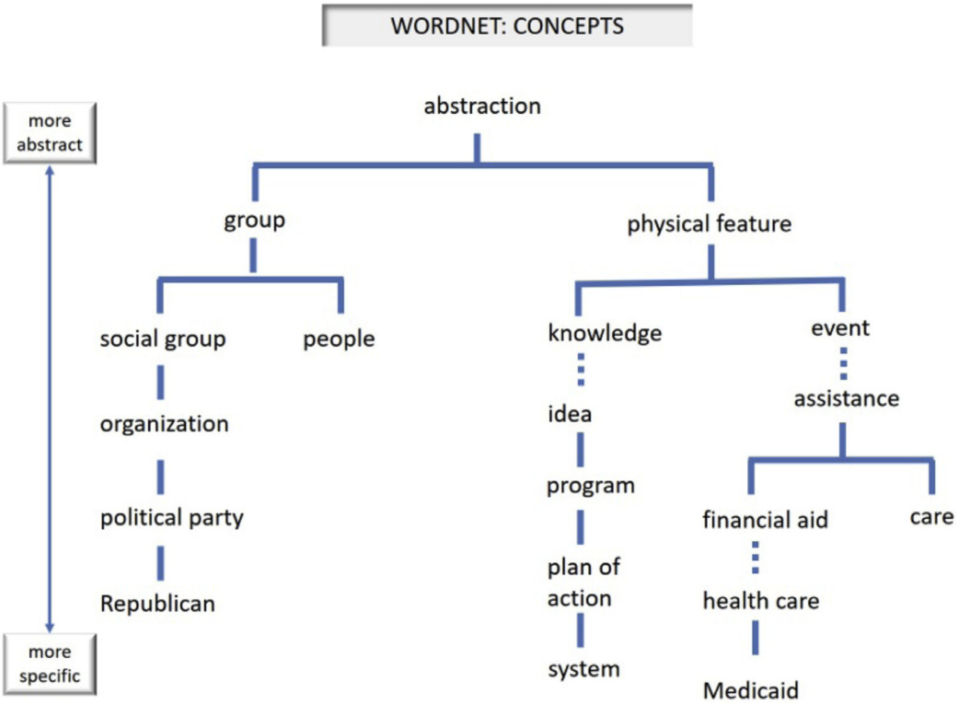

Fig. 2. Fragment of the WordNet taxonomy for some example concepts (basic concepts on leaf nodes). Solid lines represent IS-A links; dash lines indicate that some intervening nodes were omitted to save space.

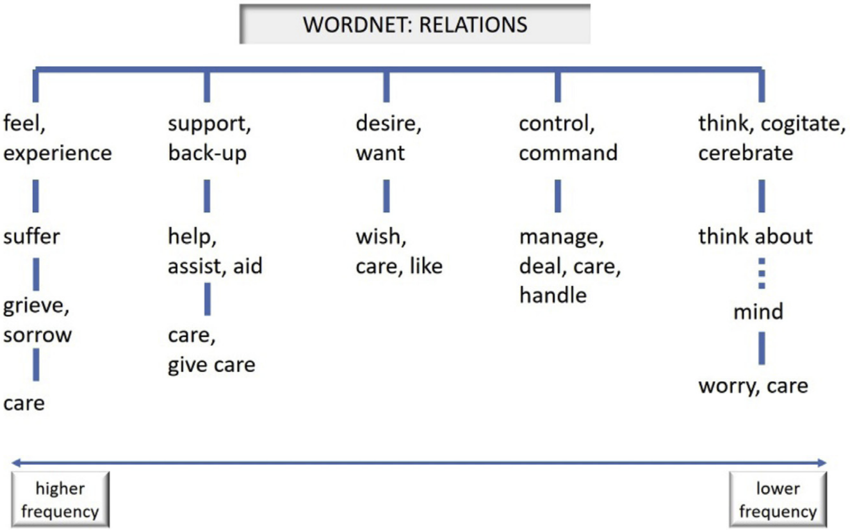

Fig. 3. Wordnet hierarchical tree for the relation "care".

r_j are semantically relevant if $dist_{rel}(r_i, r_j) < \theta$ otherwise r_i and r_j are semantically irrelevant.

For example, if we consult WordNet using the verb "*care*" we will obtain the tree shown in Fig. 3. From the tree, we observe that the list of synonyms for a specific word, are represented by the leaves, and the internal nodes are represented by the troponyms. For example, {*care, give care*} are synonyms and "*aid*" is a troponym of "*care*". Navigating through the leaves of the tree, from left to right, if r_1 and r_2 are synonymous then $dist\_rel(r_1, r_2) \leq \theta$ (relation similarity threshold), otherwise we search by tree branch and if $r_1$and r_2 are within the same branch then Eq. (5) is applied thus:

$$dist\_rel(r_1, r_2) = leng(r_1, r_2)/leng(branch) \qquad (5)$$

where $leng(r_1, r_2)$ is the length of the path between r_1 and r_2 and $leng(branch)$ is the length of the branch that contains r_1 and r_2. If no match exists for the above alternatives then $dist\_rel(r_1, r_2) = 1$. So, for example, if $\theta = 0.1$ then $dist\_rel(care,\ give\ care) =\ 0.1$ and $dist\_rel(care, aid) = 1/3$.

3.3.2. Structural similarity

Using the same notation as Section 3.1, the following gives definitions for equality, sub-meme and topological similarity.

Let $G = (V_1, E_1)$ and $H = (V_2, E_2)$ be directed graphs. V_1 and V_2 are the vertex sets of G and H respectively; E_1 and E_2 are the edge sets of G and H respectively. A homomorphism from G to H is a function f: $G \to H$, such that if u, $v \in V_1(G)$ and $(u, v) \in E_1(G)$ then $f(u)$, $f(v) \in V_2(H)$ and $(f(u), f(v)) \in E_2(H)$.

Definition 3. Equal and similar memes:
Let $M_1 = (V_1, E_1, R_1)$ and $M_2 = (V_2, E_2, R_2)$ be two memes, where V_i, E_i, R_i represent the vertex set, the edges set and the relationships set of meme M_i respectively.

- M_1 is equal to M_2 if $V_1 = V_2$ and $E_1 = E_2$ and $R_1 = R_2$.
- M_1 is a similar meme to M_2 (or viceversa) and denoted by $M_1 \approx M_2$ if a homomorphism exists between them, that is, a bijection exists that preserves the topological structure, and the concepts and relationships are mutually similar. In Definition 4 we will detail the matching mechanism required to establish meme similarity.
- M_1 is a sub-meme of M_2 and denoted by $M_1 \subseteq M_2$ if $M_1 \subseteq M$ and $M \subseteq M_2$ and $M_1 \subseteq M_2$, and M_1 is an induced subgraph of M_2.
- M_1 is a similar sub-meme of M_2 and denoted by $M_1 \sim M$ if M_1 is a sub-meme of M_2, and M_1 is an induced subgraph of M_2.

Definition 4. Similarity of a meme component: concept-relation-concept
Consider a component of a meme comprised of two concepts and a labeled relation which relates them, $M_1 = (c_1, r_1, c_2)$. Then we can compare this component to other components in which one or both of the concepts are different and/or the relation is different.

Hence, for example, the closest (identical) match could be M_1 with M_2, followed by M_1 with M_3, in which the only difference is that the relation is M_2 instead of M_1. We will use the metric discussed previously in Section 3.2 and Definition 2, to establish the distance between relations M_1 and M_2. Now consider the next matching, that of M_1 with M_4, the only difference being that the concept is M_3 instead of M. In this case, we use the metric discussed previously in Section 3.3.1 and Definition 1, to establish the distance between concepts M_2 and M_3. Now consider the matching of M_1 with M_5, in which the second concept is different and the relation is different. Hence, we will have to calculate the distance between the differing relations and the concepts. Finally, M_6 represents the case when both concepts are different and the relation is the same, and M_7 in which all three components are different.

To summarize, the minimum potential distance is represented by $dist(M_1, M_1)$ and the maximum potential distance is represented by $dist(M_1, M_7)$.

We note that in practice, as a result of the processing of real online text, multiple relations may be identified with respect to two concepts in the same meme instance. In this case, we can consider a ‘relationship set’ $R = \{r_1, r_2, \ldots r_n\}$ as existing between two concepts c_1 and c_2. Hence, one relationship set will be compared to another relationship set in order to establish the overall distance of the relationships, following the same procedure (Definition 2) repeated as for the unique relation.

3.4. Identifying the top memes

In order to identify the top memes, we use three metrics which will be defined in the following: meme frequency, meme diameter and maximum meme.

Definition 5. Frequency of a meme
The frequency of a meme is the number of times that a meme appears in the graph set M, either alone or as a similar meme to some other meme or as a sub-meme of other meme or as a similar sub-meme in a topological and semantic sense. The frequency of meme $m_{i,\,j}$ is denoted by $\mathcal{F}m(m_{i,j})$ with $m_{i,j} \in M$.

Definition 6. Minimum and maximum memes
Minimum Meme, is the meme with the minimal diameter and that is present in a maximum number of other memes. In the context of the example domain, this means a pure meme whose concepts and relationships are present in a maximum number of documents or posts. This is defined formally in Eq. (6) as follows:

$$MinMe = m_{i,j} \text{ such that } diam(m_{i,j}) \leq diam(m_{k,l}) \text{ with } \mathcal{F}r(r^{i,j}) > 0, \forall (k,l) \in I(M) \tag{6}$$

where $r^{i,\,j}$represents all relationships of meme $m_{i,\,j}$.

Maximum Meme, is the meme that has the greatest number of edges and vertices (semantic network that has the maximum semantic value). This is defined formally in Eq. (7) as follows:

$$MaxMe = m_{i,j}\text{such that } (\mathrm{i},\ \mathrm{j}) \in I((k,l)/max\{|e_{k,l}| + |v_{kl}| \forall (k,l) \in I(M)\}) \tag{7}$$

Fig. 4 summarizes the internal processing steps and parameters. The user is only required to enter the three parameters shown on the lower left: document corpus, concept frequency threshold and concept weighting. Default values are assigned for the latter two parameters and help can be given to explain the effect of the settings which can be ‘low’, ‘medium’ or ‘high’ in both cases. For expert users, an “advanced options" facility allows the user to access the internal parameters. This would be necessary, for example, if the user wishes to process a free format data corpus with a different format to general news article posts (e.g. tweets of short abbreviated texts or specialist forums with technical vocabulary). The stopword list, n-gram size and context window are similar to other text processing systems (such as Unitex) whereas the thresholds are specific to the M2TEX automatic processing funcionality (in Unitex the thresholds are defined manually). Later in Section 4 we will show how the concept frequency threshold and concept weighting are in general somewhat more sensitive to the document corpus than the other parameters.

4. Real benchmark domains – online comments forums

In this Section we apply our method which has been defined in Section 3 to process four real benchmark domains which are online text forums and the corresponding online newspaper articles, and evaluate the results. The objective will be to extract a meaningful subset of the most frequent “memes” (semantic structures represented by combinations of concept-relation-concept tuples), for each document corpus. The four chosen domains have been recently used in a study of online forums by McMillen (2013). Each domain is exemplified by its journalistic value (McMillen, 2013) and capacity to generate debate on polemic themes. In Table 2 we see a summary of their characteristics. We note that we have used all the forum posts for each dataset, whereas in McMillen (2013) only the first 200 were used. However, we have maintained the same chronological order and most recent post timestamp as in McMillen (2013). In Section 4.1 we explain the experimental setup and procedure for evaluating the assignment of the control parameters, and in Section 4.2 we present the processing results. Four different political debate themes have been chosen, from two distinct online newspapers, hence we propose (and McMillan proposed) that the results of analyzing would be reasonably generalizable to other similar forums.

4.1. Experimental setup and procedure

In order to evaluate the quality of the results we will use the well known information retrieval metrics of precision, recall and F-score. We observe that many works (see Section 2, related work) of text corpus mining use these metrics for evaluation. A key element of calculating these metrics is the definition of the “relevant set”. We do this by combining qualitative information of the key concept and relation structures using McMillen’s original study

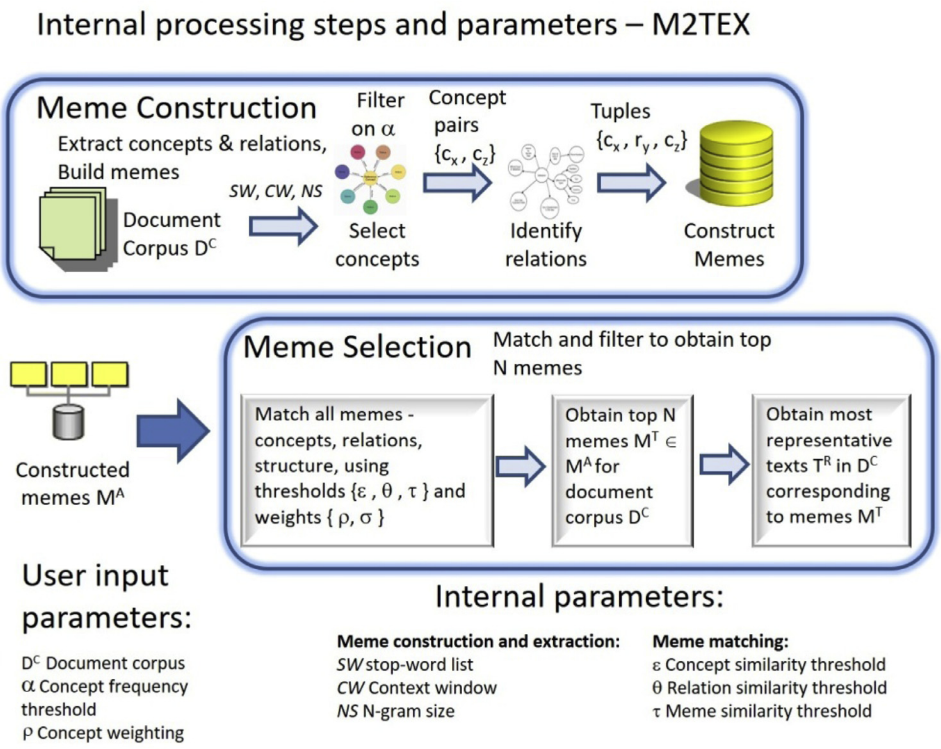
Internal processing steps and parameters – M2TEX
Meme Construction
Extract concepts & relations, Build memes
Document Corpus D^C
SW, CW, NS
Filter on α
Select concepts
Concept pairs $\{c_x, c_z\}$
Identify relations
Tuples $\{c_x, r_y, c_z\}$
Construct Memes
Constructed memes M^A
Meme Selection
Match and filter to obtain top N memes
Match all memes - concepts, relations, structure, using thresholds {ε, θ, τ} and weights {ρ, σ}
Obtain top N memes $M^T \in M^A$ for document corpus D^C
Obtain most representative texts T^R in D^C corresponding to memes M^T
User input parameters:
D^C Document corpus
α Concept frequency threshold
ρ Concept weighting
Internal parameters:
Meme construction and extraction:
SW stop-word list
CW Context window
NS N-gram size
Meme matching:
ε Concept similarity threshold
θ Relation similarity threshold
τ Meme similarity threshold

Fig. 4. Schematic diagram of the internal processing of the system and internal control parameters.

Table 2
Online forum domains used for benchmarking (McMillen, 2013).

Article name/Acronym	Theme	Number of posts in forum	Online link
Against black history month (BHM)	Black history month	229	Retrieved from http://www.nationalreview.com/articles/339649/against-black-history-month-charles-c-w-cooke
The decline of unions (DU)	Decline of unions	298	Retrieved from http://www.nationalreview.com/articles/338950/decline-unions-john-fund
Obama takes gun debate on the road: 'It's time to do something' (GD)	Gun debate	318	Retrieved from http://thehill.com/homenews/administration/280943-obama-takes-gun-control-push-on-the-road-says-its-time-to-do-something
Landscape shifts on immigration (IM)	Immigration	282	Retrieved from http://thehill.com/homenews/senate/279711-landscape-shifts-on-immigration

of the domains and our own interpretation of each document corpus, together with a quantitative evaluation of the initial results of calculating concept co-occurrence frequencies and z-scores, among other metrics. We recall from Section 3 that a meme is defined as a semantic network composed of concepts linked by relations. The system "back end" (Algorithms 1–4 and associated APIs to WordNet) was implemented in the Python programming language.

With reference to the relevant set, we use this in order to internally benchmark and calibrate (train) the system, whereas in real (production) use this set would not necessarily be available. However, given that our task is to establish parameter values which work well in general, and the effect of the two parameter values (concept frequency threshold and concept weight) which are assigned by the user can be easily seen. That is, it is easy for the user to see if the results are coherent and make spot checks on specific memes, relating them to the corresponding text posts. This scenario is typical when "calibrating" a representative "model" which is then applied and generalizes to new datasets. We validate this in Section 4.4 by "training" the parameters on one corpus and then "testing" them on the remaining corpuses.

A first task is to empirically assign values to the control parameters which give a close to optimal F-score. In order to do this we first evaluate the control parameters in process sequence to obtain an idea of the range of values which give meaningful results (by manual evaluation of the analysts together with "side knowledge" about the document corpus). By manual inspection, was evident that the best assignments for two text processing parameters remain static for different corpus, which are the n-gram size (set to 1) and the context window (set to 5). This was corroborated by testing with a different system (Unitex). Also, the relation weight is equal to one minus the concept weight so we only need to consider assigning the concept weight. This leaves us with five parameters to be evaluated by systematic statistical testing: thresholds (concept frequency, concept similarity, relation similarity, meme similarity) and concept weighting. By manual inspection, ranges and specific values were determined for each of the five parameters within which they were expected to give optimal results. These ranges were, respectively, concept frequency threshold (0.05, 0.10, 0.20), concept similarity threshold (0.1, 0.3), relation similarity threshold (0.1, 0.3), meme similarity threshold (0.1, 0.3) and concept weighting (0.60, 0.70, 0.75).

For the user interface, we used these values to assign qualitative labels to the two user input parameters, thus: concept frequency threshold (0.05 = low, 0.10 = medium, 0.20 = high) and concept weighting (0.60 = low, 0.70 = medium, 0.75 = high).

Next we used a statistical DoE (Design of Experiment) procedure to run through a set of experiments with different values assignments within the ranges defined in the first step for the five selected parameters. The assignments which gave the best F-score were then chosen. The definitions of precision, recall and F-score as given as follows in Eqs. (8)–(10).

$$\boldsymbol{Precision} = \frac{relevant\ memes\ \cap\ retrieved\ memes}{retrieved\ memes} \tag{8}$$

$$\boldsymbol{Recall} = \frac{\textbf{relevant memes} \cap \textbf{retrieved memes}}{\textbf{relevant memes}} \tag{9}$$

$$\boldsymbol{F - score} = \frac{\boldsymbol{2\,Precision \times Recall}}{\boldsymbol{Precision\ +\ Recall}} \tag{10}$$

We note that the F-score will be our target or “fitness value” which we wish to optimize. Thus our system will be calibrated to produce a result (retrieved set) which represents a tradeoff between precision and recall. In the present work the objective is to find a general control set which works quite well for all document corpus. As future work, we may evaluate finding a customized set for each document corpus.

Design of Experiments (DoE): first we consider the set of “independent variables”, which are the five input parameters: concept frequency threshold α; concept similarity threshold ε; relation similarity threshold θ; meme similarity threshold τ; weight for concepts ρ. The other two parameters are fixed as commented previously, n-gram size $NS = 1$, context window size $CW = 5$. Secondly, we consider the dependent variable which in this case will be the F-score.

Hence we wish to execute different experiments (control parameter combinations) and note the experiments which give the best F-score values. A full factorial experiment set for the number of independent variables and would give $5! = 120$). In order to reduce the search space, we used subsets (generated by a DoE statistical tool) which are calculated to give sufficient coverage to obtain a good approximation. Also, we conducted a pre-selection of candidate ranges for each parameter, by conducting ad-hoc experiments in different sub-steps, evaluating the results produced in each case. After the pre-selection, the Statgraphics software was used to generate a DoE for the given ranges.

Pre-selection of ranges for each parameter: In order to reduce the search space to find optimal values, we first conducted tests with combinations of subsets of the control parameters, where each subset controls a different part of the process.

Firstly, the n-gram size and context window size were varied to see their effect on the frequency and relevance of concepts and relations extracted. Empirically, these independent variables have a great effect on the dependent ones, and it was relatively easy to see which values gave coherent results. It was found that given the texts to be processed, an n-gram *NS* of 1 and a context window size *CW* of 5 gave the best results for all four corpuses.

Secondly, using fixed values of the parameters evaluated previously, the concept frequency threshold, and the concept, relation and meme matching (similarity) thresholds were varied to see the effect on the quality of the concept, relation and meme matches. The best results were found to occur with a value between 0.1 and 0.3, for all three similarity thresholds, and for concept frequency the best values were 0.05, 0.1 and 0.2.

Thirdly, using fixed values of the parameters evaluated previously, the concept/relation weights were varied to see the effect on the retrieved set (and thus of the precision, recall and F score). It was found that the best results were obtained by giving a preferential bias to the concepts, given that the relations were more difficult to establish, with a concept weight ρ of between 0.6 and 0.75.

The above process was repeated for each document corpus. With the ranges identified for the independent variables (control parameters), we were able to proceed to define the following DoE (Design of Experiment) using the Statgraphics software.

Statgraphics design of experiment: The Design of Experiment (using the DoE tool of the Statgraphics software, http://www.statgraphics.com/) consists of a five step process. With reference to Tables 3–5, **Step 1** consists of defining the response variables to be measured, **Step 2** the definition of the experimental factors to be varied and **Step 3** the selection of the experimental design. Then, **Step 4** consists of specifying the initial model to be fit to the experimental results. The following parameters were used: factors=” process”; model=”2-factor interactions”; “coefficients= 37. Finally, **Step 5** consists of selecting an optimal subset of the runs, from which 17 runs were selected. The optimal subset is calculated automatically by the Statgraphics DoE statistical tool. The metrics for the optimal subset were as follows: Aver. Pred. Var: 0.21, D-efficiency: 84.62, A-efficiency: 67.43, G-efficiency: 45.23. The D-efficiency was chosen as the optimization metric. It is the metric which is most generally chosen for this DoE method (Table 6).

In Table 7 we summarize the best parameters automatically found for each document corpus. Firstly, it can be seen that only two parameters vary depending on the corpus being processed: concept frequency threshold and concept weight. Corpus BHM has a slightly higher concept frequency threshold (because more concepts are identified for this corpus) and lower concept weight (as there are more concepts we can give more weight to relations). Likewise, GD can be seen to have a slightly higher concept weight (0.75) because there are less relations identified in this corpus. The results indicate that optimum processing would be obtained by adjusting the concept frequency threshold and/or concept weight for each document corpus.

Relevant set: determining the relevant set is necessary in order to obtain a realistic evaluation of the methods. As commented, this is done by domain evaluation by manual/human interpretation, which was a consensus from a group of human reviewers. Six reviewers participated in the selection of the relevant sets and the kappa (degree of consensus) statistic for their selections was 0.85.The retrieved set is then simply the raw results returned by each method. However, we have to take into account the recovery of partial memes. That is, in basic precision and recall, or you recover a document or not, it’s a binary action. Whereas in our case we can recover part of a valid meme, for example, if a meme is composed of two concepts c_1 and c_2 and a relation r_1 between them, we may recover the two concepts but not the relation. Hence, we require a partial matching for each meme and the quantitative values are summed to give the retrieved set. The relevant set will be the sum of all complete matchings for each meme in the relevant set. The sum of the precision and recall values is then normalized in order to present the results, as shown later in

Table 3
Step 1: Definition of the response variables to be measured.

Name	*Units*	*Analyze*	*Goal*	*Target*	*Impact*	*Sensitivity*	*Low*	*High*
F_score	Normalized	Mean	Maximize	1.0	3.0	Medium	0.0	1.0

Table 4
Step 2: Definition of the experimental factors to be varied.

Name	*Units*	*Type*	*Role*	*Low*	*High*	*Levels*
Concept_freq_thresh ($\boldsymbol{\alpha}$)	Number of concepts	Continuous	Controllable	0.05	0.20	3
Concept_sim_thresh ($\boldsymbol{\varepsilon}$)	Normalized	Continuous	Controllable	0.10	0.30	2
Relation_sim_thresh ($\boldsymbol{\theta}$)	Normalized	Continuous	Controllable	0.10	0.30	2
Meme_sim_thresh ($\boldsymbol{\tau}$)	Normalized	Continuous	Controllable	0.10	0.30	2
Concept_weight ($\boldsymbol{\rho}$)	Normalized	Continuous	Controllable	0.60	0.75	3

Table 5
Step 3: Selection of the experimental design.

Type of Factors	*Design Type*	*Centerpoints Per Block*	*Centerpoint Placement*	*Design is Randomized*	*Number of Replicates*	*Total Runs*	*Total Blocks*	*Error D.F.*	*Number of samples per run*
Process	Half fraction 2^8–1	5	Spaced	Yes	0	72	1	91	1

Table 6
Design of Experiment – generated combinations of independent and dependent variables (only first 6 of 49 experiments are shown for BHM corpus, for brevity).

Independent						Dependent
EXP	Concept_freq_thresh (α)	Concept_sim_thresh (ε)	Relation_sim_thresh (θ)	Meme_ sim_ thresh (τ)	Concept_ weight (ρ)	F-score
1	0.05	0.10	0.10	0.3	0.75	0.21
2	0.20	0.10	0.30	0.1	0.70	0.34
3	0.20	0.30	0.30	0.3	0.70	0.23
4	0.05	0.10	0.30	0.3	0.70	0.31
5	0.05	0.30	0.10	0.3	0.60	0.33
6	0.10	0.10	0.30	0.1	0.75	0.27

Table 7
Best values found for each document corpus.

CORPUS	N-gram_ size (*NS*)	Context_ window_ size (*CW*)	Concept_ freq_ thresh (α)	Concept_ sim_ thresh (ε)	Relation_ sim_ thresh (θ)	Meme_ sim_ thresh (τ)	Concept_ weight (ρ)	F-score
BHM	1	5	0.10	0.1	0.1	0.3	0.60	0.538
DU	1	5	0.05	0.1	0.1	0.3	0.70	0.733
GD	1	5	0.05	0.1	0.1	0.3	0.75	0.561
IM	1	5	0.05	0.1	0.1	0.3	0.70	0.564

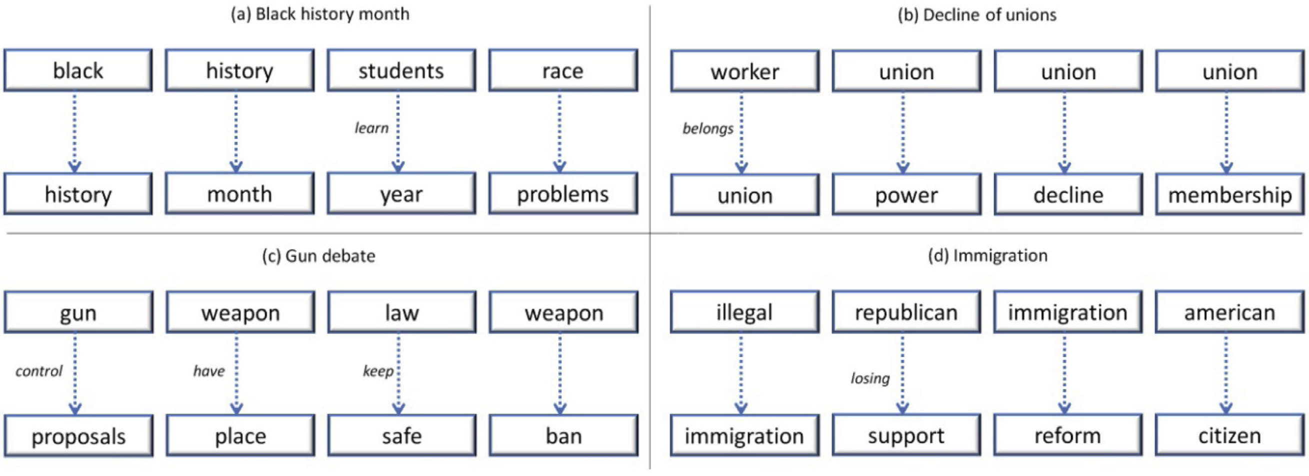

Fig. 5. Relevant sets of memes for each corpus.

Figs. 5–7. As mentioned previously, the relevant set is used for internal validation and training of the system, whereas the runtime version does not require a relevant set in order to execute. This is the general scheme employed by the calibration of a system which then generalizes to new scenarios (text corpus in this case). The user can manually inspect the results and has the option of altering the two input control parameters (concept threshold frequency and concept weight) to see their effect.

Matching of relevant set to retrieved sets: two methods have been used for matching the “relevant set” of memes with the “retrieved set” of memes, in order to calculate the precision and recall, whose basic formulas have been given previously. Essentially, *Method 1* is stricter and requires an exact matching on the concepts, summing an accumulative partial score of each tuple to give the value for the numerator part of formula for the precision and recall, that is, the intersection between the retrieved set and the relevant set. On the other hand, *Method 2* is less strict in that it allows a partial matching and then sums a 1 for each tuple match within the allowed threshold. *Method 1* and *Method 2* use Algorithms 3 and 4 as the basis for matching. The meme matching

Table 8
Processing results for articles/forums of Table 2.

Article Theme	Key concept pairs	Key relations	Key memes
Black history month	(history, month, 27), (black, history, 20), (month, year, 7), (month, history, 7), (black, white, 7), (history, people, 5), (black, people, 5), (people, people, 5), (black, white, 5), (people, white, 4), (black, race, 4), (way, year, 4), (people, right, 4)	(think, 4), (become, 4), (mix, 3), (teach, 3), (look, 3), (seem, 3), (make, 3)	#10-Meme: [(history, month, None, 19)]→[16 memes] #71-Meme: [(black, people, None, 166)]→[15 memes] #87-Meme: [(American, year, None, 216)]→[15 memes] #88-Meme: [(race, black, None, 217)]→[14 memes]
Decline of unions	(union, union, 17), (union, worker, 11), (labor, union, 7), (people, union, 6), (money, union, 6), (government, union, 6), (union, job, 6), (corporation, union, 6), (corporation, money, 6), (union, business, 5), (money, government, 5), (business, worker, 5), (government, business, 5)	(make, 12), (take, 7), (get, 7), (pay, 7), (think, 5), (go, 5), (use, 4), (force, 3), (want, 3), (realize, 2), (work, 2)	#23-Meme: [(union, business, None, 54)]→[45 memes] #74-Meme: [(union, union, get, 219)]→[44 memes] #83-Meme: [(corporation, money, make, 250)]→[35 memes] #98-Meme: [(union, union, take, 296]→[44 memes]
Gun debate	(gun, gun, 14), (Obama, gun, 11), (gun, time, 8), (time, gun, 8), (gun, Obama, 5), (gun, people, 5), (time, people, 4)), (Obama, time, 4)), (people, gun, 4), (people, people, 4)	(take, 4), (say, 3), (want, 3), (make, 3), (believe, 2), (go, 2) (protect, 2), (talk, 2)	#15-Meme: [(gun, Obama, protect, 26)]→[9 memes] #37-Meme: [(gun, time, take, 92)]→[15 memes] #38-Meme: [(gun, Obama, get, 93)]→[15 memes] #42-Meme: [(gun, Obama, talk, 248)]→[9 memes]
Immi-gration	(immigration, law, 8), (republican, party, 8), (republican, immigration, 7), (border, party, 6), (party, president, 5), (immigration, border, 5), (American, people, 5), (party, party, 4), (state, year, 4), (immigration, American, 4), (party, American, 4)	(vote, 7), (say, 6), (need, 6), (work, 4), (go, 3), (change, 3), (make, 3), (enforce, 3), (give,3), (take,3)	#35-Meme: [(state, year, None, 76)]→[12 memes] #39-Meme: [(American, immigration, None, 86)]→[11 memes] #53-Meme: [(party, president, None, 128)]→[16 memes] #88-Meme: [(republican, immigration, None, 271)]→[11 memes]

and algorithms have been described previously in Section 3 and are also detailed in the Annex.

4.2. Processing results

In this Section we will describe the key results of processing the four forums indicated in Table 2, using the processing method described in Section 3 of this paper. Table 8 shows a summary, for each article/forum, of the extracted key concepts, relations and minimal memes.

With reference to Table 8, in the last column headed ‘key memes’ we have selected the top memes, as ranked by the number of other memes which are similar to it. For example, for the document set ‘Black history month’, meme #10 is within the similarity threshold τ (calibrated to 0.3) of 16 other memes in the document set. We have validated by visual inspection that the key concepts are relevant and meaningful for the themes discussed in the corresponding texts. For example, in the case of the “Immigration” document set, the concept pairs {immigration, law}; {Republican, immigration}; {immigration, border}; and {immigration, american}, among others, capture essential aspects of the current debate about immigration in the United States. The relations, on the other hand, are more subjective and disperse than the concepts, and their meaning is less apparent when they are taken out of their contextual text embodiment. The key memes which are most frequent do not have relations assigned (see Fig. 5). This is partially due to the weighting values which biased concepts ($\rho = 0.7$) over relations ($\sigma = 0.3$), because we considered it is more essential to have both concepts with assigned values (not null) over a relation without a value. This is inevitably results in a trade-off, which can be specified by the user.

With reference to Table 9 we have included the corresponding posts for each meme. Note that in the definition of Meme #23, the number 54 refers to post (or document) 54 for the corresponding corpus (Decline of Unions in this case). Therefore, it is easy to index the post corresponding to the meme and show it to the user. In this way we make the output of the system closer to Pingar than Unitex (see descriptions of these system in Sections 2 and 5) Table 9 gives representative examples of the resulting output which are shown to the user. Each post is selected as the most representative to the meme which embodies the core themes (and the most frequently) being discussed. Also, we also identify the 45 memes which are most similar to Meme #23 (shown between brackets in the meme definition). The Meme identifiers are not enumerated for brevity, however these are available and also index the corresponding posts so we can show a second level of summarization (a set of the most similar posts to the most representative post).

As next step of the user output, Table 10 shows the top memes which correspond to Meme #25 of Table 9, and their corresponding posts. Thus the user can (optionally) see the key meme and its list of most similar memes through which the original meme is propagated. It can be easily seen that the posts follow the theme of business, corporations and the unions.

In Fig. 5 the top memesfrom each relevant set of memes for each corpus are shown. These sets are used as the reference set for the precision and recall calculations. They are defined by human experts by manual inspection of the articles and corresponding forum posts together with “side knowledge” of the frequencies, z-scores, minimal and most frequent memes As mentioned previously, the forum posts have many cases of grammatical incorrectness, semantic or logical incoherencies, reference to a topic (concept) as an article without explicitly naming it, among other issues.

Thresholds: we recall that the frequency threshold α is used for extracting concepts (see Section 3.3.1). In Table 11 selected results of empirical tests are shown for different values of α, such as 0.05, 0.10 and 0.20. We recall that α represents the acceptable percentage frequency which an *n-gram* must appear in order to be considered a concept. In Table 11, for example, we interpret that

Table 9
Posts corresponding to top memes shown in Table 8 for Decline of the Unions corpus.

#23-Meme: [(union, business, None, **54**)] → [45 memes]
- **Post 54:** Over the last five decades I've noticed three groups, that have done more damage to the Unions in this country, then anyone else. Number One is the EPA, starting with, and continuing till today, with the Clean Air act. It virtually destroyed manufacturing Steel, and their Unions. Secondly the Democrat Party, which has been in full support of any Federal Regulation, and taxes aimed at Manufacturing in this Country. Number Three, Union Leadership, who attack productivity, and efficientcy in what ever business they deal with, mostly manufacturing.

#74-Meme: [(union, union, get, **219**)] → [44 memes]
- **Post 219:** You are kidding I presume. We outperformed the world until we unionized teachers. Now, sadly we languish behind much of the Western world, while listening to the union liars say it is "for the children." How many children get raises after the strike? How many children have their political campaigns funded by the Union slush fund? How many children get cushy union pensions? They feel they must remind us that they are "selfless public servants" with their picket signs full of grammatical errors.

#83-Meme: [(corporation, money, make, **250**)] → [35 memes]
- **Post 250:** Please, think. Greed, absent political cronyism, never got a corporation one single dime. The only way a corporation can make money is if they have a product or service the public is willing to buy... ...unless... They get in bed with corrupt politicians who can award contracts their way or rig the rules to favor them. There's a very easy solution: Get government the heck out of most business, and into protecting property rights and freedoms.

#98-Meme: [(union, union, take, **296**] → [44 memes]
- **Post 296:** You don't need a Republican Congress. All it will take is a Republican President and an Executive Order. Kennedy used the EO to permit public sector unions so it just needs a President with the cahones to take on the Union.

Table 10
Most similar memes to Meme #23 in Table 9 and their corresponding posts.

#25-Meme: [(union, business, None, 54)] -> [70,72,81,95,...]

70 #70-Meme: [(money, business, make, **205**)]
- **Post 205:** Business men like to make money. They do that by making good products at the lowest possible price. The only way to do that is to hire the best. You can't do that if you are paying less than your competitors are paying. The price of labor will always approach it's marginal utility. Push it below that point, and your best workers leave. Push it above that point, and the company goes out of business. Anyone who tells you that the stock market is a good indicator of the economy, is lying to you.

72 #72-Meme: [(job, job, lose, **210**)]
- **Post 210:** Only a complete fool believes that low costs leads to no jobs. Out here in the real world, those who save money, spend that money on other things. It takes people to make those other things. There are no jobs lost, just new jobs elsewhere in the economy.

81 #81-Meme: [(money, people, allow, **224**)]
- **Post 224:** You misunderstand. It's not corporations per se that are the source of evil. It's money. More specifically, people being allowed to have more money than xxx believes they are entitled to.

95 #95-Meme: [(corporation, people, think, **277**)]
- **Post 277:** Perhaps. But that's not the point. Government has no business bailing out anyone. And businesses cannot force people to buy their product the way unions can force companies to hire union workers.

Table 11
Values used for the concept similarity threshold and corresponding percentage of concepts included for different document sets.

Document set	Concept frequency threshold (α)		
	0.05	0.10	0.20
Black history month	11.45	22.9	45.8
Decline of unions	14.9	29.8	59.6
Gun debate	15.9	31.8	63.6
Immigration	14.1	28.2	56.4

for the 'Black history month' corpus and $\alpha = 0.10$, 22.9 of all available *n-grams* were assigned as concepts.

In the next step (see Section 3.3.1) we take all the concepts whose frequency is equal to or greater than α, and use these to find the relations. Then, in the meme matching (see Section 3.3.2), weights ρ and σ, where $\rho + \sigma = 1$, are use to indicate the relative importance of concepts and relations respectively. From the best values found in the DoE (see Section 4.1), ρ and σ are assigned the values 0.7 and 0.3, respectively, thus giving a bias on the concepts with respect to the relations. We recall that in order to match a meme, at the lowest level of detail we are matching concepts and relations. Concepts are matched using the distance calculation as defined in Definition 1 of Section 3.3.1), with the assigned optimum similarity threshold $\varepsilon = 0.1$. Relations are matched using the distance calculation as defined in Definition 2 of Section 3.3.1), with the assigned relevance threshold $\theta = 0.1$. Also, at a higher level of comparison, the threshold τ is used for defining the overall distance for two memes to be considered similar. After empirical testing, the value of τ was assigned as 0.3, given that for higher values some inconsistencies were detected in the similarities between the memes.

Some concept pairs may not seem evident at face value. For example, in the case of the 'Gun Debate' document set, the concept 'time' seems to have a disproportionate relative importance. However, if we look at the original article we see that the title is "Obama takes gun debate on the road: 'It's time to do something'". If we read through the comments we see that many contributors have picked up on this phrase, repeating it or using it to make 'plays on words' . The sense of the phrase is that now is the moment to deal with this issue, due to the increasing availability of high powered/automatic firearms and assault weapons to the general public. The following are some example comments:

Obama is correct: it's time to do SOMETHING.

It is time for the President his friends to quit taking money from the biggest profiteers from gun violence.

Time to start respecting the constitution.

Time to stop putting future generations into debt.

Time for a new president, then anything else.

We' ve had plenty of time hearing the lunatic side of the story. It is time for Congress to act.

However 'time' is also used with other meanings, such as in the following comment:

A gun in the home makes the likelihood of homicide three times higher, suicide three to five times higher, and accidental death four times higher. For every time a gun in the home injures or kills in self-defence.

In Fig. 6a and b the precision and recall values are shown for each document corpus and the different concept frequency threshold α values. The α values range from 0.05 to 0.2, where a higher threshold reduces the number of selected concepts, based on fre-

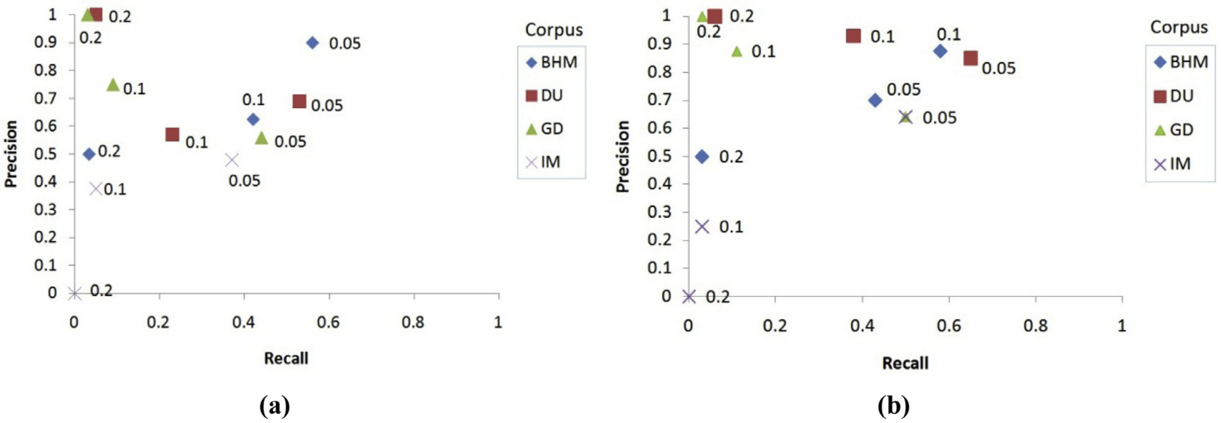

Fig. 6. Precision and recall for different document corpuses and concept frequency thresholds α (0.05, 0.1, 0.2): (a) method 1, (b) method 2.

quency in the document corpus. As described previously in the Design of Experiment, two methods have been used to matching the “relevant set” of memes with the “retrieved set” of memes, in order to calculate the precision and recall: *Method 1* (requires exact matching) and *Method 2* (allows partial matching). The results of applying methods 1 and 2 are shown in Fig. 6a and b, respectively.

From Fig. 6a it can be seen that the threshold α of 0.05 gives the best overall equilibrium between precision and recall, followed by 0.1. A threshold of 0.2 gives significantly worse results because the “retrieved set” is very small. In terms of the corpus we see that for a threshold of 0.05, corpus BHM gives the best results, followed by corpus DU. For threshold 0.1, again corpus 1 gives the best results followed by corpus 2. This is again essentially due to the size of the “retrieved set”. In terms of absolute values of precision and recall, we see that the optimum area ($\alpha = 0.05$) in the graph is delimited with recall = 0.37→0.56 and precision 0.55→0.9, giving the best equilibrium between precision and recall. In terms of the F-score, this corresponds to a range of 0.42 → 0.69 for the four document corpus.

In Fig. 6b (method 2) we see a similar scenario to Fig. 6a (method 1) with respect to the threshold α, where a value of 0.05 gives the best overall equilibrium between precision and recall. Also, corpus BHM and DU again show the best results. However, in terms of the absolute values of precision and recall, we now see a higher density of points in an area of the graph is now displaced upwards (better recall) and to the right (better precision). That is, the delimited area corresponds approx. to recall = 0.43→0.65 and precision 0.64→0.85, giving the best equilibrium between precision and recall. . In terms of the F-score, this corresponds to a range of 0.53 → 0.74 for the four document corpus.

Hence, in conclusion, we observe that the threshold of $\alpha = 0.05$ gives the best overall results for M2TEX applied to the four corpuses. We note that the threshold α has been taken as an example to illustrate the evaluation made for the other control parameters through the Design of Experiments described in Section 4.1. Later in Section 5 of the paper we will apply a set of benchmark methods to the same corpuses and compare the results with those of M2TEX.

4.3. Benchmarking of computational cost

In this Section we evaluate the computational cost of the two main components of the processing, semantic network formation and meme selection. It can be seen that meme selection is the process which occupies the majority of processing time. In order to benchmark the scalability we have replicated the corpuses up to 20 times, that is we have processed the original corpus, the original corpus × 2, …, the original corpus × 20. Fig. 7a and b show the benchmark results for the two main processes, “semantic network construction" and “meme selection", for which the average values have been taken for all corpus. As expected, the most computationally intensive process is the meme selection. In Fig. 7a and b we see that both trend lines fit to a second order polynomial, in which the semantic network construction is less linear ($R^2 = 0.84$) as the corpus size increments, whereas in Fig. 7b we see that the meme selection time has a tighter fit to the second order polynomial ($R^2 = 0.99$). For typical forums sizes we propose the processing time is adequate, and the most computationally intensive process, the meme selection is that which fits closer to linear. For processing multiple forums a parallel version could be developed as future work, and both processes could be further analyzed for optimization.

The system was developed using a PC with Intel® Pentium® Dual CPU T3400 processor with a speed of 2.16 GHz, RAM, 3GB of RAM, Windows 7 operating system (32 bits) without threading. The code was programmed in Python, without using any external libraries and is interpreted by IDLE (Python GUI).

4.4. Processing results - generalization

In Section 4.1 we tested and compared the results of M2TEX on each forum, in terms of optimization of the parameters to produce an F-score. In this Section we test M2TEX’s capacity to generalize, that is, calibrating the system parameters on a given forum and then testing them on another to see how the F-score (which is the fitness measure) behaves. For example, we train on BHM and then test on DU, GD and IM. Next we train on DU and test on BHM, GD and IM, and thus for the remaining two forums. The results are summarized in Table 12.

We found that the F-measure maintains relatively stable for all combinations, however, some forums have different frequencies and distributions of concepts and relations so it can be concluded that a customized assignment of two of the parameters (concept frequency threshold and concept weight) over the four forums would give best overall performance. This agrees with the parameter setting results detailed in Section 4.1 (Table 7).

Table 12 shows the result of setting the optimum parameters on one document corpus and testing on the remaining three, in each case. It can be seen that the parameters give similar optimum values and the parameters for one corpus generalize to the other three corpuses. For example, assigning the parameters on BHM and then testing on BHM gives an F-score of 0.54. Then the assigned BHM parameter settings were used to test on DU, GD and IM, which gave F-scores of 0.51, 0.45 and 0.49, respectively.

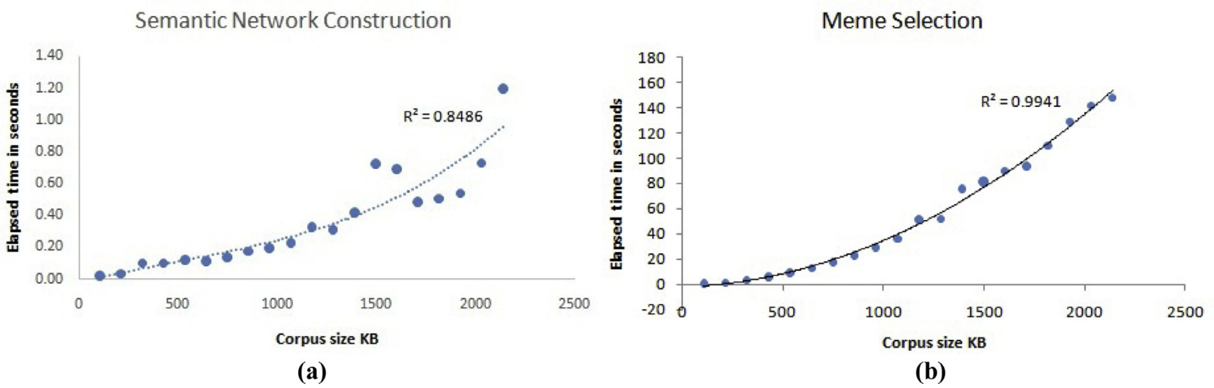

Fig. 7. Execution time for (a) semantic network construction and (b) meme selection.

Table 12
Cross tests between document sets (F-score).

Calibrated on	Tested on			
	BHM	DU	GD	IM
BMH	0.54	0.51	0.45	0.49
DU	0.51	0.73	0.53	0.53
GD	0.45	0.53	0.56	0.51
IM	0.49	0.53	0.51	0.56
Average	*0.50*	*0.58*	*0.51*	*0.52*
Standard deviation	*0.04*	*0.10*	*0.05*	*0.03*

From the parameter setting results that were given in Table 7, it would be expected that the parameter settings of BHM and GD would be relatively less optimal for the other forums. In Table 7, corpus BHM had a concept frequency threshold of 0.10 as opposed to 0.05 for other three corpuses, and a concept weight of 0.6 whereas the average was 0.72 for other three corpuses. Also, corpus GD had a concept weight setting of 0.75 whereas other corpuses had settings between 0.6 and 0.7.

From the statistics in Table 12, it can be seen that the average values of the F-score are still competitive with the other metrics and systems benchmarked (see results in Tables 13 and 14). Also, the standard deviations of the F-score are small (between 0.03 and 0.05), with the exception of DU, which is slightly higher at 0.10. In conclusion, it can be seen that the parameter settings generalize well between corpuses, although, as was expected, the optimum processing is obtained by varing the two user control parameters (concept frequency threshold and concept weight).

5. Benchmarking with other systems

In order to evaluate the performance of our proposed method, we will use two systems, Pingar (Medelyan & Divoli, 2012) and Unitex (Muniz et al., 2005) that are widely used robust industry standard and academic tools. They have been described previously in the state of the art. They provide contrasting results, given that Pingar uses keyword frequency and density to automatically identify key text blocks which may contain co-occurrences. On the other hand, the metrics used from Unitex identify co-occurrences directly, and calculate the frequency and z-score statistics. Also we have manually derived a set of memes for each corpus using the z-score results of Unitex. We will call these methods *Pingar, Unitex all* (memes + frequency based + z-score based), *Unitex freq., Unitex z-score* and *Unitex memes*. Our method will be referred to as M2TEX. Finally, we have carried out a user survey in which volunteers punctuate the top tuples extracted by the different methods for each corpus. Whereas en Section 4 we used two methods to match the "relevant set" of memes with the "retrieved set" of memes, in order to calculate the precision and recall, in this Section we apply only *Method 2* (allows partial matching). See Section 4.1 (Design of Experiments) for a description of each matching method.

5.1. Processing procedure: Pingar

We have applied Pingar to the four document corpus used in Section 4, applying the document summarization API (*api-demo.pingar.com*) in which the number of desired paragraphs is given as a parameter, as well as a query search term. Pingar then selects a reduced number of paragraphs from the complete document corpus which it considers to be the most representative according to their weight in the whole document. It also extracts a list of keywords that could be single words or compound noun phrases which it considers the most significant. Next, by manual inspection, we can identify the location of the keywords in the summarized texts. From these we can identify co-occurrences of keywords and finally any candidate relations (such as verbs) that are positioned between the co-occurrences. For brevity and due to space restrictions in the paper, we have edited the resulting text paragraphs, excluding repetitions of the same concepts and co-occurrences.

The user interface has three control parameters: summary length (number of paragraphs), query term (optional), number of

Table 13
Processing results for all methods and corpuses "black history" and "union decline".

	Black history			Union decline		
	Precision	Recall	F-score	Precision	Recall	F-score
M2TEX	0.700	0.437	0.538	0.846	0.647	0.733
Unitex -all	0.722	0.217	0.333	0.667	0.607	0.635
Unitex z-score	0.875	0.117	0.206	0.375	0.088	0.143
Unitex freq	0.750	0.100	0.176	1.000	0.235	0.381
Unitex mem	0.000	0.000	0.000	0.625	0.147	0.238
Pingar	0.417	0.083	0.139	0.250	0.059	0.095

Table 14
Processing results for all methods and corpuses “gun control” and “immigration”.

	Gun Control			Immigration		
	Precision	Recall	F-score	Precision	Recall	F-score
M2TEX	0.64	0.5	0.561	0.646	0.5	0.564
Unitex -all	0.714	0.313	0.434	0.500	0.210	0.295
Unitex z-score	0.200	0.031	0.054	0.300	0.048	0.083
Unitex freq	1.000	0.156	0.270	1.000	0.161	0.278
Unitex mem	1.000	0.125	0.222	0.000	0.000	0.000
Pingar	0.571	0.125	0.205	0.583	0.113	0.189

keywords (optional). By default, the summary length is set to 2 and the number of keywords is set to 5. After testing with different values the default values were found to give the best result. Note that the end-result of processing by Pingar is a text summary where named entities are explicitly highlighted, but not concepts or relations.

5.2. Processing procedure: Unitex

Unitex (http://www-igm.univ-mlv.fr/~unitex/) is a text analysis open-source tool, which mainly employs grammar rules and dictionaries for the analysis. It has been widely utilized in research, teaching and industry and relies on Finite State technologies. Unitex incorporates large-scale language resources for several languages:, including English, Portuguese and French. It allows the detection of relations among text tokens, the incorporation of dictionaries and searching for patterns through regular expressions set by the tool.

We have applied Unitex to the four document corpus used in Section 4. Unitex is a system which can be made to work more closely (than Pingar) to our concept and relation matching methods (see Definitions 1 and 2 of Section 3.3.1). First we use it to identify a set of key concepts. Secondly, for each key concept, we use it to find the top co-occurrences by frequency and by z-score. **Co-occurrences are identified by using a window of five tokens to the left and to the right of a given key concept**. Finally, we search for frequent verbs which are present between the top co-occurrences. We must highlight that this was a process in which each step had to be executed manually. For instance, the corpus document had to be preprocessed by the use of another system (Lucene) in order to apply stemming and eliminate stopwords because the Unitex preprocessing step only does tokenization. After doing this previous step it was more effective to calculate the word frequencies to select the top values. This contrasts with our system in which the process is automatic.

The Z-score measure provides a calculated value that shows how far and in what direction a pair of words deviates from the distribution’s mean. The calculation is expressed in terms of the distribution’s standard deviation. It presents a graduation among word pairs which largely corresponded with the semantic collocation frequencies expected. To summarize, the z-score calculates the relevance of two terms in a co-location.

For the search funcionality of ’collocations’ of Unitex, the only control parameter offered by the system is the token window which defines how many tokens to the left and right of a target concept are considered in order to find a concept pair.After trying different window sizes, a window of 5 tokens to the left and 5 tokens to the left was found to give the best results and coincides with the optimum window range of ±5 found for our system M2TEX. Given 5 tokens to the left and 5 to the right with respect to the word in focus we are providing the length of the context of the word in order to find strong collocations. If the range is set to 3 tokens, the window to find collocation of two words becomes too limited; on the other hand, if we set it to 9, the length of the context that involves the word becomes too permissive, including the analysis of tokens that are far from the word in focus. Thus, the search of collocations using a window of 5 gave the best results.

5.3. Comparison of processing results: M2TEX, Pingar and Unitex metrics

Our system (M2TEX) automatically extracts semantic networks from a free text corpus, where the semantic networks represent the essential and most trended topics. It can be seen from the results that the Pingar and Unitex systems require manual processing in order to identify the top concept co-occurrences and relations. Comparing the results of our system with Pingar and Unitex, the most frequent key concepts were quite similar, although we understand that the focus of Pingar is on "entities" and “key phrases” rather than single concepts in a more general sense (e.g. “Harry Reid” and “Chicago”). We also understand that Pingar chooses a chunk of text because it has a high relative "density" of keywords with high weight, so in that chunk there could be some co-occurrences which aren’t necessarily the most frequent throughout the whole corpus.

On the other hand, M2TEX is frequency based, which means that it misses out some less frequent but interesting combinations which were identified by Unitex using the “z-score" instead of simple frequency, such as {border, patrol} in the case of the “immigration” corpus and {heritage, month} in the case of the “black history month” corpus. However, M2TEX is designed to identify “meme” type high frequency combinations which propagate through a network such as an online forum, hence we insist that our priority on frequency is correct.

Each of the three systems tested has been developed with different functional requirements to fulfil, but as the aim of the comparison is to measure the reliability of our system at the moment of extracting key concepts and relations, we can see that our system extracts representative or meaningful concepts and relations which were also identified by Pingar to make a summarization or by Unitex after the non automatic intervention needed.

In Fig 8a scatter plot is shown of precision and recall for all methods and corpuses. M2TEX is distributed in a central region of the plot whereas in general the other methods are distribution closer to the x-axis (lower recall). The highest precisions were obtained by the Unitex frequencies and memes, however at the cost of a low recall. In Fig. 9 the average values (over the 4 corpuses) are shown for the precision, recall and F-score for each method. It can be seen that M2TEX has the best equilibrium between precision and recall. “Unitex all” has a slightly better precision but a lower recall.

In Tables 13 and 14 the average precision and recall are detailed for each method and corpus. It can be seen that the scores are corpus and method dependent. For example, M2TEX scores lower on precision but higher on recall for “black history”, than for the other three corpuses, whereas Unitex-all scores higher on precision for “black history” with respect to the other corpuses, but lower on recall.

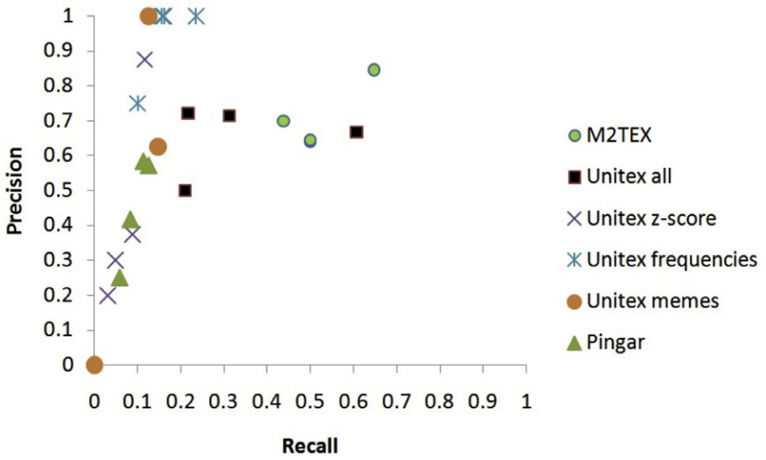

Fig. 8. Precision and recall for each document corpus and compared method.

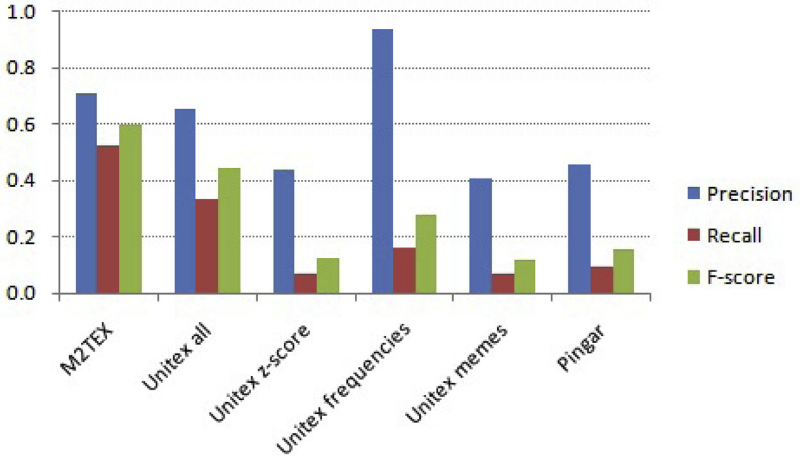

Fig. 9. Precision, recall and F-score for all methods, aggregated by corpus.

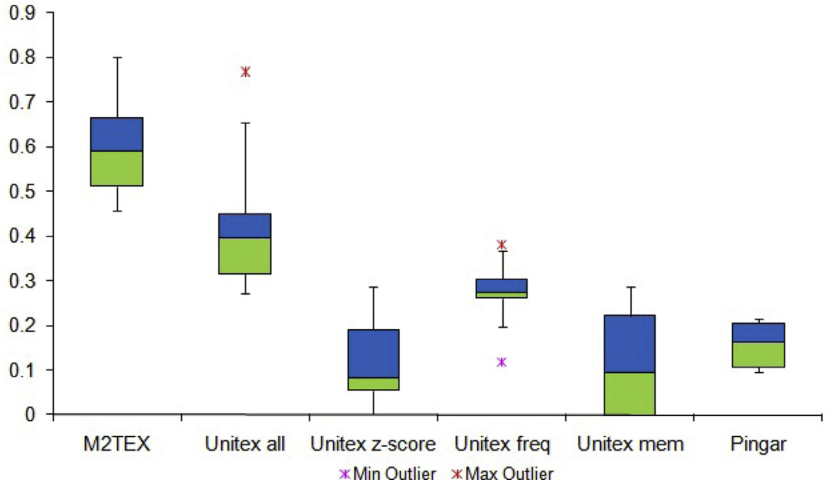

Fig. 10. Box-and-Whisker plot of F-score for all methods.

Fig. 10 shows a box plot of the F-scores for each method, which are calculated based on the disaggregated values for each meme tuple (top 2), corpus and method. For each method the box plot shows the second (green) and third (blue) quartiles of the values, and the whiskers show the first (lower) quartile and the fourth (upper) quartile. In this way we can see the distribution and range of the values for each method. In Fig. 10 it can be seen that M2TEX has the range which is displaced higher towards the maximum possible F-score, followed by “Unitex all”. The worst performers were “Unitex z-score” and “Unitex mem”, due to their low Recall values. The “whiskers” show the standard deviation from the mean and it can be seen that M2TEX and “Unitex all” have the widest

Table 15
False positive statistics (% of retrieved).

Corpus	System		
	M2TEX	Unitex	Pingar
BHM	0.32	0.22	0.12
DU	0.23	0.50	0.65
GC	0.40	0.73	0.61
IM	0.29	0.23	0.50
Average:	0.31	0.42	0.47

standard deviations, where the majority of F-scores tend to fall in the fourth quartile, which indicates a more favorable result.

Table 15 shows the statistics for false positives. We see that M2TEX has a lower count than Unitex for DU and GC and a lower count than Pingar for DU, GC and IM. Note that Unitex produces a list of collocations and the remainder of the process performed automatically by M2TEX was done manually for Unitex. The same applies for Pingar which produces a summarized text with named entities indicated. We also note that as partial matching is applied for all results: a returned result whose partial matching was not superior to 0.3 for any relevant result was considered to be a false positive.

As commented in Johnson et al. (2006), one source of false positives are words with multiple meanings. Incorporating term synonymy matching resulted in a lower 'correctness', ranging from 42% to 94%, with an average of 67.4%. Also, as discussed in Manning, Raghavan, and Schütze (2008), even for good systems, trying to label more documents as relevant will often lead to a higher false positive rate. However, having a high non-relevance threshold and thus labelling many documents as nonrelevant (and as a consequence reducing the false positive rate) results in an unsatisfactory user experience. Hence, the system should be designed to have a balance so that it produces documents for the user to see, with a certain tolerance for showing some false positives as long as the retrieved set provides some useful information to the user.

5.3.1. Manual evaluation of output of each method

Up to now we have used statistical methods and metrics to evaluate the results. In order to contrast this approach with a manual evaluation we enrolled a group of 15 volunteers to perform a "blind" evaluation of the output of each method for each corpus. That is, each volunteer evaluated the output using a set of qualitative criteria, without knowing the method to which the output belonged (they were referred to as 'method 1', method 2', and so on). Each volunteer was asked to read the summary of each corpus (the original article), and then grade the output of each method for each corpus, using three subjective criteria, coverage, relevance and informativeness. The grade used a scale of 1 to 5, where 1 = bad, 2 = insatisfactory, 3 = average, 4 = satisfactory and 5 = good.

In Table 16 the results are summarized, from which it can be seen that the overall top method is Manual (the gold standard), followed by M2TEX and Unitex-all. In terms of individual criteria, M2TEX was strongest in 'coverage', whereas it came second in 'relevance' and 'informativeness' (after Manual). It is recalled (from Section 4.2) that 'Manual' represents a set defined by human experts for each corpus by manual inspection of the articles and corresponding forum posts together with "side knowledge" of the frequencies, z-scores, minimal and most frequent memes.

We note that the volunteers evaluated the tuple sets with respect to the article alone, as the volunteers could not be expected to read all the forum posts. Hence, we interpret that the volunteers have given a lower punctuation to the tuple sets for all methods with respect to the article, because they have not compensated for the additional difficulty of including and interpreting the corresponding forum posts. However, we defend this approach because all methods were evaluated under the same conditions and hence all methods were 'scored down' in this context and in the same manner.

5.4. Processing of article vs. comment posts

In this paper, the focus is on extracting semantic structures from free format text, meaning the article together with the following comment posts, comparing methods for extracting meme like structures. Hence a detailed comparison of results of processing the article versus processing the comments posts is out of the scope of the current work, and it will be tackled in a future project. However, the results do enable us to reach conclusions with respect to processing the article and comments forums together. Processing the article alone is a significantly easier task, because it is well written and focused on a specific issue. However, in the case of the comment posts there is no guarantee that the users will keep 'on concept' or will write their posts well (that it, grammatically, spelling, comprehensibility, etc.). Thus, processing the comment posts is a significantly bigger challenge for identifying and extracting semantic type structures which repeat throughout the body of posts for a given article.

In general, we have found that for the four articles and comment forums processed, the key concepts extracted do tend to coincide with the original article. In the case of the gun debate and immigration articles, there was a certain concept drift into a political centric debate between Republicans and Democrats. Conversely, specific named entities from the article in the gun debate (e.g. Senator Harry Reid) were not repeated/referenced in the posts. Some specific aspects were focussed on more in the posts (e.g. assault weapons) than in the original article.

In terms of the quality of knowledge extraction, it is clear that the more badly written and ad-hoc posts texts produced tuples with concept pairs but without a corresponding relation. This can be because the writer did not explicitly define the relation in his/her post, abbreviated in reply to a previous post or to the article itself, or did not construct a correct grammatical structure or argument, among other aspects. One improvement with respect to this problem would be to complete the partial constructions using "side information" (e.g. the original article), however this would run the risk of introducing misinterpretations. In order to minimize misinterpretations, a probabilistic calculation could be made of the most likely relation in a given context.

6. Discussion and conclusions

In this paper we have presented a system for meme processing (extraction, construction and comparison), using information retrieval and semantic network concepts. The data processing procedures have been applied to four extensive real online forums and corresponding articles, in order to test and validate the method. We have then benchmarked the same document corpuses with a set of other methods in order to corroborate and compare the extraction of the concepts, relations and co-occurrences. The results confirm that our system was successful in automatically extracting key knowledge (memes) which has both relevance and coverage, from unstructured free format online text.

The implemented system combines a semantic network extractor with a semantic network matcher which is able to process unstructured text and identify the key representative themes within the text. By designing a system which works well with default control parameters, an end user (text analyst) without expert knowledge (about natural language processing) or an expert user who wishes to automate part of the analysis and processing steps, will be able to use it to extract knowledge from online text corpuses, which can be customized for specific end-user requirements. To

Table 16
Manual evaluation of the output of each method.

	Coverage		Relevance		Informativeness		Overall
	Avg.*	St.Dev.*	Avg.	St.Dev.	Avg.	St.Dev.	(C+R+I)
Unitex -all	3.03	0.61	2.89	0.73	2.64	0.96	8.56
Unitex meme	1.83	0.56	2.14	0.67	1.97	0.96	5.94
Pingar	2.39	0.55	2.56	0.62	2.36	0.96	7.31
M2TEX	3.50	0.59	3.42	0.79	3.28	0.96	10.19
Unitex freq	2.11	0.82	2.14	1.13	1.81	0.96	6.06
Unitex z-score	1.92	0.79	2.14	0.91	1.97	0.96	6.03
Manual	3.17	0.65	3.71	1.04	3.50	0.96	10.38

* Average and standard deviation for all participants and corpus.

the best of our knowledge, there are no systems currently available which provide this combination of functionality.

A key 'managerial insight' demonstrated in Sections 4 and 5 is that our method M2TEX, which incorporates similarity matching at several levels (concept, relation, semantic structure), is able to automatically process unstructured comments post forums without a pre-tagging or machine learning phase, and without human intervention between steps. This is one of the motivations of the work: to substitute a costly manual revision of hundreds of comment posts for online articles and produce an automatic summary of the key thematic information being discussed in a semantic structure form.

The research contributions in terms of expert systems and intelligent systems, as we outlined in the introduction are the creation of an integrated system which allows the user (we suppose a text analyst) to experiment in extracting memes (most representative semantic networks) from unstructured online document corpus (such as forums associated with online content). The user can use either default values or assign custom parameters in the case of expert users. The system incorporates different intelligent components, such as the matching algorithms and rule set definitions for the text extraction and threshold values.

A key contribution of our method is to be able to give a balanced precision and recall even when processing unstructured, badly written texts, for which other methods had a more severe reduction in recall. M2TEX uses different thresholds to match concepts, relations and calculate a similarity value for semantic structures, which has a frequency component and a compactness component in order to find the most repeated 'compact' memes in an article text and corresponding comments post forum. All of this is done in an automated sequence, without intermediate human intervention between steps (e.g. manual tagging), or the need for the text to be in a structured form, which is characteristic of other systems (see State of the Art).

With respect to expert and intelligent systems, an expert system embodies a set of rules and heuristics which represent human and machine expert knowledge. In the current work knowledge is embodied as rules to control (i) the concept/relation extraction and meme construction and (ii) the matching process. In the first case, the rules can be customized by the end user (data analyst) depending on the data s/he wishes to extract and the characteristics of the text corpus being analyzed. For example, an automatic extraction mode will find the most frequent memes, whereas if specific keywords are used to filter concepts or relations, the system could be used as a query based information retrieval tool, however this is out of the scope of the present work.

Hence the expert knowledge is embodied throughout the system using rule sets and heuristic algorithms. Also, the distance metric can give different levels of importance to the concepts, relations and meme similarity using corresponding weights. They can be assigned by defining an experiment set with a DoE (Design of Experiment) statistical method. A stochastic method (such as simulated annealing) could be considered to find the optimum value for the document corpus being processed. However, this was not considered in the current work due to the high computational cost (compared to the DoE approach which successfully found a competitive parameter set combination with respect to the other methods) and given that we wished to facilitate the inspection of partial results for different parameter combinations.

Strengths and weaknesses, advantages and limitations: as mentioned previously, our method is able to process unstructured text in an automated sequence and produce semantic structures from the text which represent the key knowledge themes which are present. Many other methods (see Section 2) require the text to have a structured form (XML, predefined sections, or correctly written) and/or require human intervention between steps (for example, labelling for supervised learning). However, our method does require calibration of the different thresholds for concept, relation and semantic structure mapping. Although default values worked quite well for all four corpuses, the number of concepts and relations extracted is corpus dependent, and fine tuning will give optimum results. This may be done by experimentation (i.e. using a design of experiment (DoE) approach for testing the different parameter combinations) and/or using an optimization function to find a global minimum error, with respect to a reference set of "relevant memes". As was mentioned at the end of Section 2 when we discussed the relation of the method with the state of the art, a limitation of M2TEX is that it does not explicitly calculate the meme metrics of 'longevity', 'fecundity' and 'copy-fidelity as the final step of the process (Heylighen & Chielens, 2013). However, these characteristics are implicitly considered when we perform the matching phase to choose the top memes.

As future work and research directions we can consider the evaluation of concept drift in the comments forum with respect to the original article, and the implementation of an automatic calibration mechanism for the thresholds which could use a Monte Carlo type method for finding the optimum values (i.e. using a Gaussian function to generate candidate values). Furthermore, the comments forum could be "mined" for the presence of novelty; that is, using information retrieval metrics to identify new knowledge which is relevant to the key concepts but which is not currently in the "relevant set".

Acknowledgements

This research is partially supported by the Spanish MEC (HIPERGRAPH TIN2009-14560-C03-01). The authors are grateful to Dr. Horacio Saggion of the TALN Research Group, Univ. Pompeu Fabra, for his advice on the usage of the Unitex system.

Algorithm 2 : Extraction of memes and relationships.

```
Input: Set of documents $D \neq \emptyset$, set of concepts $C \neq \emptyset$;
Output: Set of memes $M$, array of relationships $R$;
Process:
let $M = MS \leftarrow \emptyset$;
for each document $d_i \in D$ /* Relationship extraction */
    if $|d_i \cap C| = 1$ then /* the document has a unique concept */
      let $k$ be the index of such concept and let $ms_{i,1} \leftarrow \{(c_k)\}$; .
    else {
        let $j \leftarrow 1$ and $ms_{i,j} \leftarrow \emptyset$;
       for each pair of concepts $(c_k, c_l)$ such that $\{c_k, c_l\} \in d_i$ and $c_k \neq c_l$
          if $c_k$ and $c_l$ are concepts related then verify that such relationship is "normal" or "superset" and put $(c_k, c_l,$ relation_type$)$ into $ms_{i,j}$;
       let $j \leftarrow j + 1$; let $MS \leftarrow MS \cup ms_{i,j}$}
 for each $ms_{i,j} \in MS$ /* Calculation of “Closest Superset” */
      for each $c_k \in C \cap ms_{i,j}$ {
      let $CS(c_k) \leftarrow \emptyset$;
      for each $c_l \in C \cap ms_{i,j}$ with $c_k \neq c_l$ Build superset $(c_k)$ }
 for each $ms_{i,j} \in MS$ { /* Constructing minimal semantic network or meme */
      let $m_{i,j} \leftarrow \emptyset$;
      sort the concepts, in ascending order, according to the size of $d_q(c_k)$ with $c_k \in ms_{i,j}$; let $OC$ be this ordered set;
      for each $c_k \in OC$
         Find, from $k+1$ to $n$, the first relationship type “Superset” and put $(c_k, c_l,$ Superset$)$ into $m_{i,j}$;
      $M \leftarrow M \cup \{m_{i,j}\}$ }
    for each $m_{i,j} \in M$ /* Extraction of Relationships */
       if $|m_{i,j}| = 1$ then let $c_k$ be such element and $r^{i,j}_{k,0} = E$; /* empty relationship */
       else
          for each tuple $(c_k, c_l, r^{i,j}_{k,l}) \in m_{i,j}$
            Identify each relation associated with a specific concept pair, load into $R$;
 EndProcess
```

Annex

In the following the pseudo code of two algorithms is given which are related to the text processing descriptions given in Section 3 of the paper. Algorithm 2 implements concept and relation extraction and meme construction, whereas Algorithm 3 implements meme matching for finding the top memes for each corpus.

In Algorithm 3, the variable *tm* indicates the type of matching between the two memes. Recall that each meme is a labeled directed graph on the vertices (concepts) and on the edges (relationships), so if m_i is a meme then $m_i = \{(c^i_k, c^i_l, r^i_{k,l}), (c^i_l, c^i_m, r^i_{l,m}), \cdots\}$ where c^i_k is the k^{th} concept of the i$^{\text{th}}$ meme and $r^i_{k,l}$ is the relationship between the concepts c^i_k and c^i_l. In order to calculate the distance between either two concepts or two relationships we use the same method referred to in Section 3 of this paper. For Algorithm 3, line 7, ρ and σ are weights, for which, $\rho + \sigma = 1$, that indicate the relative importance we give to the concepts and the relationships between them, respectively. In line 9, $\tau \in (0, 1)$ is a threshold which represents the minimum distance in order for two memes to be considered similar.

Algorithm 3 : Meme matching.

```
/*  Uses distmeme function (embodied in Algorithm 4) to calculate the $tm$ variable.*/
1.  Input: memes $m_1$ and $m_2$, weights $\rho$, $\sigma$ and thresholds $\varepsilon$, $\theta$ and $\tau$;
2.  Output: $tm$ /* type of matching between the two memes */
3.  Process:
4.  let $tm \leftarrow 0$;
5.  sort $m_1$ and $m_2$ according to the first component of the tuples and subsequently according to the second component;
6.  if $|\mathbf{m}_1| = |\mathbf{m}_2|$ then
7.     $dist(m_1, m_2) \leftarrow distmeme(m_1, m_2, \rho, \sigma, \varepsilon, \theta)$;
8.     if $dist(m_1, m_2) = 0$ then $tm =$ ‘equal’;
9.     else if $0 < dist(m_1, m_2) < \tau$ then $tm =$ ‘similar’;
10.  else
11.    if $|\mathbf{m}_1| < |\mathbf{m}_2|$ then {
12.       for each $i \in \{1, \cdots, |m_2| - |m_1| + 1\}$
13.          let $d_p(i) \leftarrow distmeme(m_1, \{(c^2_k, c^2_l, r^2_{k,l})_i, \cdots, (c^2_k, c^2_l, r^2_{k,l})_{i+|m_1|-1}\}, \rho, \sigma, \varepsilon, \theta)$;
14.       let $dist(m_1, m_2) \leftarrow min_{i \in \{1, \cdots, |m_2|-|m_1|+1\}}\{d_p(i)\}$;
15.       if $dist(m_1, m_2) = 0$ then $tm =$ ′$m_1$ submeme of $m_2$′;
16.       else if $0 < dist(m_1, m_2) < \tau$ then $tm = m_1$ similar submeme of $m_2$’ }
17.  else { /* $|\mathbf{m}_1| > |\mathbf{m}_2|$ */
18.       for each $i \in \{1, \cdots, |m_1| - |m_2| + 1\}$
19.       let $d_p(i) \leftarrow distmeme(\{(c^1_k, c^1_l, r^1_{k,l})_i, \cdots, (c^1_k, c^1_l, r^1_{k,l})_{i+|m_2|-1}\}, m_2, \rho, \sigma, \theta, )$;
20        let $dist(m_1, m_2) \leftarrow min_{i \in \{1, \cdots, |m_1|-|m_2|+1\}}\{d_p(i)\}$;
21.       if $dist(m_1, m_2) = 0$ then $tm =$ ′$m_2$ submeme of $m_1$’;
22.        else if $0 < dist(m_1, m_2) < \tau$ then $tm =$ ‘$m_2$ similar submeme of $m_1$’}
23.  EndProcess.
```

Notes: In the methodology described in Section 3.3, the objective of the meme matching (Algorithms 3 and 4) is to identify similar memes and count their frequency, thus identifying recurrent minimal-compact memes which represent the key ideas circulating in a given document corpus. That is, to identify the most compact and minimal semantic networks (memes) which occur with the greatest frequency in a given document set.

Algorithm 4 : Meme distance, return the distance between two memes of equal cardinality*.

```
1. Input: memes $m_1 = \{(c_k^1, c_l^1, r_{k,l}^1), (c_l^1, c_m^1, r_{l,m}^1), \cdots\}$ and
          $m_2 = \{(c_k^2, c_l^2, r_{k,l}^2), (c_l^2, c_m^2, r_{l,m}^2), \cdots\}$
          , weights $\rho$ and $\sigma$, thresholds $\varepsilon$ and $\theta$
2. Comment: The distances between concepts "$dist\_con(c_k^1, c_k^2)$" and
   between relationships "$dist\_rel(r_{k,l}^1, r_{k,l}^2)$" are calculated according to
   the procedures given in Section 3.3.
3. Output: distmeme
4. Process:
5. Assign distance between memes as:
   $distmeme \leftarrow$
       $\rho(\sum_{k \in first\_comp}^{|m_1|} dist\_con(c_k^1, c_k^2, \varepsilon) + \sum_{l \in second\_comp}^{|m_1|} dist\_con(c_l^1, c_l^2, \varepsilon))/2 +$
           $\sigma \sum_{k,l}^{|m_1|} dist\_rel(r_{k,l}^1, r_{k,l}^2, \theta)$;
6. return distmeme;
```

* Note: Cardinality of a meme. It is understood that a meme is a graph (semantic network), and that we apply the standard definition for graphs: "two graphs have the same cardinality if they have the same number of vertices and links". Algorithm 4 is a subprogram of Algorithm 3 (see annex at end of paper). Algorithm 4 takes two graphs of the same cardinality (same number of vertices and links) and determines their semantic distance. Algorithm 3 consider three possibles cases of cardinality: |m1|=|m2|, |m1| ⟨ |m2| and |m1| ⟩ |m2|) given that a meme can be included within another (hence the notion of submeme). The algorithm is not restrictive, given that if the cardinalities are distinct the distance of the smaller meme is calculated for all possible matches with the larger meme, and the smallest distance is returned.

References

Banko, M., Cafarella, M. J., Soderland, M., Broadhead, M., & Etziono, O. (2007). Open information extraction from the Web. In *Proceeding IJCAI 2007* (pp. 2670–2676).

Baydin, A. G., & López de Mántaras, R. (2012). Evolution of ideas: A novel memetic algorithm based on semantic networks. In *Proceeding of IEEE congress on evolutionary computation, CEC 2012, IEEE world congress on comp. intelligence* (pp. 2653–2660). WCCI. 2012.

Beck-Fernández, H., & Nettleton, D. F. (2013). Identification and extraction of memes represented as semantic networks from free text online forums. *Proceeding of 10th international conference on modeling decisions for artificial intelligence*.

Bordogna, G., & Pasi, G. (2013). A fuzzy approach to the conceptual identification of ememes on the blogosphere. In *FUZZ-IEEE 2013, Proceeding of IEEE international conference on fuzzy systems, Hyderabad, India, 7–10 July 2013* (pp. 1–8).

Chen, Z., Gangopadhyay, A., Karabatis, G., McGuire, M. P., & Welty, C. (2007). Semantic integration and knowledge discovery for environmental research. *Journal of Database Management, 18*(2007), 43–68.

Davies, M. (2010). The corpus of contemporary American english as the first reliable monitor corpus of english. *Literary and Linguistic Computing (LLC), 25*, 447–464.

Dawkins, R. (1989). *The selfish gene* (2nd Ed.). Oxford University Press.

Heylighen, F., & Chielens, K. (2013). In R. A. Meyers (Ed.), *Cultural evolution and memetics, article prepared for the encyclopedia of complexity and systems science, 2013*.

Jean-Mary, Y. R., Shironoshita, E. P., & Kabuka, M. R. (2009). Ontology matching with semantic verification. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 7*(3), 235–251 (2009).

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceeding of international conference on research on computational linguistics (ROCLING X)* (pp. 19–33).

Johnson, H. L., Bretonnel Cohen, K., Baumgartner, J. R., W. A., Lu, Z., Bada, M., et al. (2006). Evaluation of lexical methods for detecting relationships between concepts from multiple ontologies. *Pacific Symposium on Biocomputing*, 28–39.

Kallipolitis, L., Karpis, V., & Karali, I. (2012). Semantic search in the World News domain using automatically extracted metadata files. *Knowledge-Based Systems, 27*, 38–50.

Kok, S., & Domingos, P. (2008). Extracting semantic networks from text via relational clustering. In *Proceedings of ECML/PKDD* (pp. 624–639).

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge discovery and data mining (KDD '09)* (pp. 497–506).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

McMillen, S. R. (2013). *Threads of deliberation: A textual analysis of online news comments* Master of Science thesis. United States: Scripps College of Communication of Ohio University https://etd.ohiolink.edu/ap/10?0::NO:10:P10_ACCESSION_NUM:ohiou1368025601..

Medelyan, A., & Divoli, A. (2012). Mining unstructured data: practical applications. *Pingar. O'Rielly Strata Conference, February 28–March 1 2012,* http://strataconf.com/strata2012/public/schedule/detail/22499.

Mooney, R. J., & Nahm, U. Y. (2003). Text mining with information extraction. In *Proceedings of 4th international MIDP colloquium, multilingualism and electronic language management, September 2003* (pp. 141–160).

Muniz, M. C. M., Nunes, M. G. V., & Laporte, E. (2005). UNITEX-PB, a set of flexible language resources for Brazilian Portuguese. In *Proceedings XXV congresso da sociedade brasileira de computaçao, July 2005* (pp. 2059–2068).

Nettleton, D. F. (2013). Data mining of social networks represented as graphs. *Computer Science Review, 7*, 1–34 February 2013.

Oh, J., Kim, T., Park, S., Yu, H., & Lee, Y. H. (2013). Efficient semantic network construction with application to PubMed search. *Knowledge-Based Systems, 39*, 185–193 (2013).

Peng, M., Gao, B., Zhu, J., Huang, J., Yuan, M., & Li, F. (2016). High quality information extraction and query-oriented summarization for automatic query-reply in social network. *Expert Systems with Applications, 44*(February), 92–101.

Resnik, P. (1999). Semantic Similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research, 11*, 95–130.

Ross, S. (1976). *A first course in probability* (9th ed. 2014). Pearson Education Limited.

Rubio, L., De la Sen, M., Longstaff, A. P., & Fletcher, S. (2013). Model-based expert system to automatically adapt linning forces in pareto optimal multi-objective working points. *Expert systems with Applications, 40*(2013), 2312–2322.

Simmons, M. P., Adamic, L. A., & Adar, E. (2011). Memes online: Extracted, subtracted, injected, and recollected. In *Proceedings ICWSM, 2011*. The AAAI Press.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of 2nd international conference on information and knowledge management* (pp. 67–74).

Szumlanski, S., & Gomez, F. (2010). Automatically acquiring a semantic network of related concepts. In *Proceedings of 19th international conference on information and knowledge management* (pp. 19–28).

Thangaraj, M., & Sujatha, G. (2014). An architectural design for effective information retrieval in semantic web. *Expert Systems with Applications, 41*(December(18)), 8225–8233.

WordNet. (2010). *About wordnet*. Princeton University http://wordnet.princeton.edu.