Zeszyty Naukowe Wydziału Informatyki
Wyższej Szkoły Informatyki Stosowanej i Zarządzania
„Informatyka Stosowana”
Nr 1/2014

TOPIC DETECTION AND TRACKING: A FOCUSED SURVEY AND A NEW VARIANT

Marek Gajewski*, Janusz Kacprzyk*,, Sławomir Zadrożny***

* Systems Research Institute, Polish Academy of Sciences, Warszawa,
ul. Newelska 6, 01-447 Warszawa, Poland

** Warsaw School of Information Technology,
ul. Newelska 6, 01-447 Warszawa, Poland

Abstract

A theoretically challenging and practically important problem of information retrieval (IR), known as topic detection and tracking (TDT), is considered. Its origins and a standard formulation are briefly reminded and recast in a version suitable for our discussion. It is discussed in a broad context of IR and its relevant specific task of text categorization (TC). Moreover, a nonstandard relevant practical problem of the classification of textual documents is presented, discussed and confronted with both the TDT and TC problems.

Keywords: information retrieval, topic detection and tracking, text categorization

1. Introduction

Information retrieval (IR) is as a branch of computer science dealing with textual information processing. It has a long tradition and a well-established repertoire of tools and techniques to represent and process collections of documents (here and later on by a document we will mean a textual document). A fundamental problem considered in IR is how to effectively and efficiently retrieve documents that are relevant for a particular user who expresses his or her information need in the form of a *query*. A number of comprehensive solutions to this problem has been proposed which are known as *models of information retrieval*. Basic classical models include the Boolean, vector space and probabilistic one. They constitute starting points for a plethora of other approaches differing in the way the documents and queries are represented and matched.

The first approaches to the representation of textual information were based on *keywords*, i.e., individual words or phrases which are identified as important for conveying the content of a document. Various models proposed differ in the way keywords are extracted from documents and how they are combined to represent a document in the process called *indexing*. This basic form of a document representation proved to be effective and efficient, and also relatively easy to implement, and has been widely used in many information systems for years.

However, some weaknesses of such a keyword based representation have been quickly recognized. The most important is the difference between the vocabularies used by the authors of various documents and the vocabularies of the users searching for the documents. Thus, for example, the use of synonyms may cause very similar documents look very different when the keywords extracted from them are compared literally. The same, of course, applies to queries. In fact, in case of queries the problem is even more evident. A collection of documents may be indexed in a consistent way by a group of experts or automatically. However, a query is often formed by a casual user who may be not aware of the vocabulary used in the collection of queried documents. Another problem are, e.g., homonyms which in a naïve keyword based approach may cause the retrieval of irrelevant documents due to an identical lexical form and a completely different meaning of two occurrences of a homonym as in a "bank [deposit]" versus a "[river] bank". There are many techniques which mitigate these types of problems like the use of *thesauri*, lexical databases listing synonyms and other semantically related keywords for a given keyword, or *word sense disambiguation* algorithms which help to identify the meaning of a homonym based on its occurrence context. However, more sophisticated text representation methods seems to be a more effective solution to the mentioned problems.

These advanced representations may be characterized in the simplest way as referring to the *concepts* alluded to in a document instead of just keywords. That is, they refer to the "real" semantics of a document, not just of a word. An automatic indexing of a collection of documents with the use of concepts is obviously more difficult than in the case of keywords. In an extreme case, it requires "understanding" the content of a document and then selecting proper concepts reflecting its meaning, i.e., mimicking the way a human indexer works.

However, other successful techniques have been elaborated which do not require such a deep semantical analysis of a document which is still beyond the capabilities of modern information processing systems. They are basically exploiting the statistics of the co-occurrence of keywords in a collection of documents. Such co-occurring groups of keywords are then treated as representing some higher level concepts. These concepts may be of a different nature: completely opaque as, e.g., in case of the Latent Semantic Indexing (LSI) (Letsche and Berry, 1997) or explicitly referring to the underlying keywords as, e.g., in case of the Latent Dirichlet Analysis (LDA) (Blei, 2003) where concepts (known therein as *topics*) are probability distributions over the space of keywords.

Having at hand a representation of documents, be it a simple, keyword-based, or a more sophisticated, e.g., referring to the concepts, many other tasks related to textual information processing may be dealt with. Among them, a prominent role is played by an automatic classification of documents to a set of predefined categories, usually referred to as *text categorization* (TC) (Sebastiani, 1999, 2002, 2005). The categories under consideration may be distinguished in many different ways and due to that many practical problems may be cast in the context of text categorization. In this paper we discuss one of such practical problems, namely the *topic detection and tracking* (TDT) (Allan, 2002), and another related problem which we have recently defined (Zadrożny et al., 2013). We will mention their specifics, importance, and challenges related with their formulation, analysis and practical use.

The paper is organized as follows. In the next section we briefly recall the basics of the problem of text categorization (TC). In Section 3, the topic detection and tracking (TDT) problem is discussed. The main approaches proposed in the literature to solve the TDT problem are also briefly summarized. In Section 4, a recently defined problem of TC, namely of the classification of textual documents to sequences of documents is presented and its similarities and differences with respect to the TDT are discussed. We conclude the paper with some concluding remarks on the state of the art, challenges, the solution proposed, future research directions, etc.

2. The text categorization problem

The task of textual documents classification may be understood in two basic ways. First, it may be identified as a task of clustering, i.e., grouping documents in such a way that similar, in some sense depending on the considered application, documents fall into the same groups (clusters) and documents from different groups are not similar. This is a very important task, in particular when there are not known any classes which may be reasonably assigned to the documents. Thus, the goal is to discover such a grouping based on characteristic features of the collection of textual documents under consideration. Any of the unsupervised learning techniques may be applied, notably all clustering algorithms such as the hierarchical clustering, *k*-means or Kohonen's Self Organizing Maps (Everitt et al., 2010). In fact, clustering was among the first specific techniques used to deal with collections of documents (Salton, 1971). The initial application was meant primarily to preprocess a document collection so as to provide for a faster execution of queries. Recently, it has been frequently proposed to postprocess results of query execution so as to group returned documents and thus make the output of a query more comprehensible to the human user. Clustering also plays an important role as a tool to solve the TDT problem as it will be discussed later on.

Second, it may be assumed that there is given a set of classes and the task is to assign all documents to one or more of them. Often, these classes are referred to as *categories* and the whole process is called *text categorization* which is convenient as one can avoid a possible ambiguity in distinguishing this task with the previous one, of the clustering type. Both the clustering and categorization techniques may be employed together to solve practical problems such as the TDT problem which is considered in this paper.

The research on the text categorization problem has a long tradition. The problem may be illustrated with the task of assigning volumes in a library to genres such as, e.g., romance, detective story, thriller, guide, dictionary, etc. The grouping of books according to the genres makes life of both a librarian and library customer easier as he or she may faster deliver or find a book of interest.

This example immediately shows two aspects of text categorization. Namely, the categories often form hierarchies, e.g., one can distinguish two top level categories such as “fiction” and “non-fiction”. Both of them may be partitioned into subcategories which in turn may be further partitioned into susubcategories, and so on. Another aspect of this text categorization scenario is related to the character of categories. In the example under consideration, the categories are distinguished based on the thematic content of the books. Many other examples in this vein may be given: newswire stories may be grouped into categories such as politics, economy, sports, etc., web documents served by a site may be grouped based on their main topics, etc.

This is however just one possible interpretation of categories. Many different practical problems may be cast as a text categorization task and the understanding of the concept of a category may be different in their particular frameworks and settings. For example, classes of books may be identified with their authors, i.e., each class comprises books authored (or co-authored) by a given person. Then, the principle of categorization does not refer to the content of a book but rather to its metadata (the name of the author, in this case). Another example concerns the grouping of publications according to their type: a book, an article, a chapter in an edited volume, or a paper published in conference proceedings. We will come back later to the important issue of the very nature of a category.

Whatever the character of a category is, the following notation may be used to more formally analyze the problem of text categorization:

- $D = \{d_1,...,d_n\}$ is a set of documents, (1)
- $C = \{c_1,...,c_m\}$ is a set of categories of documents. (2)

The process of text categorization may be executed in several ways. However, its basic form consists in a direct assignment of a category to a document by an expert involved in the process. Thus, such an assignment is based on the judgment of a human being as to the belongingness of a document to a category which is assumed to be possible to carry out, i.e., a function:

$$A: D \to C \tag{3}$$

is assumed to exist.

A special case of (3) of a particular practical importance is when C contains just two elements. A prominent example is here the task of documents filtering (Baeza-Yates and Ribeiro-Neto, 2011) where these two categories correspond to relevant/irrelevant or interesting/uninteresting pairs of notions.

On the other hand, a more general form of function A in (3) takes the form:

$$A: D \to 2^C \tag{4}$$

which assumes that a set of categories may be assigned to a document. Such a more general formulation of the problem is referred to as the *multilabel text categorization*.

Of course, if a constraint is imposed on A such that hat $A(d)$ is a singleton set, then the original setting defined by (3) is recovered. It is worth noticing that (4) encompasses also the possibility that $A(d) = \emptyset$, i.e., a classifier represented by (4) is allowed not to assign any category to a document which is relevant for our further considerations of the TDT problem.

A solution to the text categorization problem depending on an expert making a classification decision for each document is however impractical in the case of a large volume of documents to be classified which is typical in the case of, e.g., the Internet resources.

Hence, some automated approaches have to be applied. Two types of such approaches may be conceived. Both consist in forming a broadly meant set of rules to classify/categorize documents and differ in how this set is arrived at. The first type belongs to the realm of *knowledge engineering* and assumes the rules are hand-crafted and form a basis of a kind of an expert system which is then used to automatically classify documents. An example of such an approach is the CONSTRUE/TIS system (Hayes and Weinstein, 1990) used in the past by the Reuters company to categorize newswire stories. Solutions of this type reduce the burden of "manual" categorization of documents. However, still a huge effort has to be put in the development and maintenance of such an expert system mentioned above.

The second type of approaches, belonging to the realm of *machine learning*, makes it possible to replace knowledge engineers with some automatic means to establish a classifier ("set of rules") based on a training dataset comprising documents with assigned categories. Thus, in this case one of the multitude of supervised learning algorithms may be employed such as the decision trees, support vector machines or artificial neural networks, to name just a few (Sebastiani, 2002; Yang and Liu, 1999). A caveat consists in the required availability of a usually quite large training dataset of a good quality. Anyway, this type of approaches is studied in the literature most extensively and is of our primary interest also in this paper.

An important aspect of the text categorization problem is the mode in which documents are to be categorized. Namely, the whole collection of documents to be classified may be available at once which makes it possible to exploit some statistics for the whole collection. Another mode boils down to the *on-line categorization* when documents are considered one-by-one and are immediately categorized individually.

An in-between mode is also sometimes considered where documents for classification are presented to the system in bulks (portions) of a fixed or changing size. This last mode, of course, covers the two previous ones when the size of bulk equals the size of the whole collection or 1, respectively.

The same modes may be considered in the context of the documents clustering. Then, the on-line mode is even more challenging as in the initial stage the system has to group documents possessing just information on a document which has just arrived and a few documents seen earlier.

3. The topic detection and tracking problem

3.1 The origins and the definition of the problem

Topic Detection and Tracking (TDT) was a part of the DARPA Translingual Information Detection, Extraction, and Summarization (TIDES) program. It was also closely related to the initiative known as The Text REtrieval Conference (TREC). The data sets used in the TDT related contests and experiments are still available via the Linguistic Data Consortium (LDC). Research on the TDT started with a pilot study in 1997 (Allan et al., 1998) and was followed by regular workshops during the next seven years.

The problem of Topic Detection and Tracking (TDT) consists in classifying incoming newswire and broadcast *stories* (documents) into groups concerning the same *topics* (categories). The stories are assumed to be coming from different sources and there may be multiple stories referring to the same topic/event at the same moment of time or evolving over some time period. The main difference from the basic text categorization problem discussed in the previous section is that these topics are not known in advance. Thus, even if the process may start with some documents assigned to some predefined categories, it has to be assumed that documents belonging to yet unknown categories may appear on the input. Such documents have to be properly recognized and assigned to a new, established for them, category. As we will see, the TDT problem then calls for the application of a combination of a text document clustering and categorization techniques. Another distinguishing aspect of the TDT problem is the fact that the documents are assumed to be time stamped. This information may be employed in particular when documents on a given topic are expected to be produced at some point in time, i.e., there are much higher chances that a given document belongs to a recently established topic (category) than to the one which was detected a long time ago.

In the original version of the TDT problem (Allan et al., 1998) the topics were identified with *events*, such as a specific volcano eruption at some place in the world or particular parliamentary elections in a given country. Later on, a more elaborate terminology and understanding of particular notions of the TDT evolved. Still, there is no widely accepted unified terminology in use. The following basic concepts may be, however, distinguished:

- *a story* is a single document/newswire story conveying some information to the user; in general, a stream of stories on the input is assumed which are more or less explicitly separated one from another;
- *an event* is something what happens at a particular place and a particular time;
- *a topic* is an important event considered together with all related events.

In what follows we will be mostly concerned with the concepts of a story and of a topic, corresponding in the basic text categorization problem to a document and a category, respectively. Thus, we will denote the set of all stories as D and the set of all topics as C, preserving the notation adopted for the text categorization, cf., (1)-(2). However, it should be emphasized that the concept of the TDT topic is somehow more specific than the concept of the category considered in the text categorization tasks. We should emphasize here that it is important an event may trigger a new topic. For example, stories on two different earthquakes will be often treated as concerning the same subject (i.e., earthquakes) but from the point of view of the TDT they will be usually seen as separate topics. Moreover, the set of topics C is not known in advance here in contrast to the set of categories in the case of text categorization. Thus, the set of topics is more like a set of clusters which has to be yet discovered in the text document clustering problem.

The following subtasks may be recognized in the TDT problem (Allan et al., 1998):

- *segmentation*, i.e., the separation of individual stories from the input stream; often it is assumed that the input stream is a transcription of an audio input and distinguishing particular stories is a non-trivial problem itself; in this paper we will not consider this subtask assuming that the input is a sequence/stream of clearly separated textual documents;
- *topic detection*, i.e., the recognition of all topics appearing in a corpus of stories D or, equivalently, the grouping of all stories into an initially unknown set of topics C; it may be considered in the off-line version, i.e., when all stories are available before the grouping starts – it is then known as *retrospective topic detection*, or in the on-line version, i.e., when stories are available one after another in the input stream and the grouping has to be carried out incrementally after the analysis of each story; the latter version is much more important for the practical purposes and the former was considered mainly in the beginning of the TDT research;
- *first story detection (FSD),* i.e., the recognition if an incoming story belongs to an already known topic or should initialize a new topic when stories are considered in the on-line mode, one by one; it also known as the *on-line new event detection*; the first story detection task may be seen as a part of the topic detection subtask but is usually distinguished and considered as a separate task; a hierarchical variant of both the topic detection and first story detection is also considered and known as the *hierarchical topic detection* (HDT) (Allan et al., 2003) in which a hierarchy of topics is assumed and particular stories may belong to many topics at different levels of a hierarchy;
- *topic tracking*, i.e., the classification of new stories to earlier discovered topics; in its basic form it assumes that a (small) set, $D_c \subseteq D$, of stories belonging to one topic $c \in C$ is known by the system and the incoming stories are to be judged as belonging to the same topic or not; basically, the stories classified by the system are not taken into account when subsequent incoming stories are classified, i.e., the set D_c in its original form is used all the time but a variant of the topic tracking problem, known as an *unsupervised adaptive tracking task,* is also considered when stories judged by the system as belonging to c are then added to D_c and thus may influence the judgment concerning subsequent stories;

- *link detection,* i.e., the deciding if two given stories belong to the same topic; this task, although for sure very important as a part of other above mentioned tasks, has not enjoyed a broad interest as a separate task among researchers dealing with the TDT.

The tasks mentioned above are closely related one to another but are considered separately to focus on their specific difficulties and to test the algorithms proposed to solve them using specific evaluation measures.

3.2 Methods proposed in the literature

For all of the above mentioned subtasks a number of solutions has been proposed in the literature. Let us briefly remind some of those proposed for the *topic detection*, the *first story detection* and the *topic tracking* problems.

Already during the TDT Pilot Study (Allan et al., 1998) at the end of the 1990s, a set of interesting approaches was developed by such leading research, academic and commercial US institutions as DARPA, Carnegie Mellon University, Dragon Systems, or the University of Massachusetts at Amherst. Those approaches had been improved later on and new ones are still being proposed in the literature. We will now briefly discuss the main techniques proposed so far.

The stories are represented using the standard IR techniques, mostly within the framework of the vector space model. Document processing steps such as the stopword elimination and stemming are usually applied. Some approaches employ more sophisticated representation schemes, going beyond the simple keywords, e.g., via the use of concepts meant as in the *topic modelling*, cf., e.g., (Blei et al., 2003). Also the use of *named entities* is reported as enhancing the effectiveness of implementation of various TDT tasks (Kumaran and Allan, 2005).

The *retrospective topic detection* variant basically boils down to the problem of clustering the whole corpus of documents (stories) given at once. Stories are usually represented using the classic vector space model. Distance/similarity measures used for the clustering purposes may take into account the lexical features of stories (keywords/terms weights) as well as their time stamps. The use of the latter is motivated by the assumption that the longer distance in time between two stories the lower the chance that they concern the same topic (event).

As the stories similarity measure often the cosine of the angle between their vector representations is employed which is among the most popular classic metrics used in various IR applications. As the number of topics (clusters) cannot be preset, then the clustering algorithm have to discover this number automatically. One of possible solutions (Allan et al., 1998) is to cluster stories incrementally in the following simple way. Each existing cluster C_k is represented by its *centroid*, $\overrightarrow{c_k}$, and a story *d* to be assigned to a cluster is compared with the centroids of all clusters. If the highest similarity $sim = \max_k similarity(d, \overrightarrow{c_k})$ exceeds a preset threshold value, *thres*, then this story is assigned to the cluster C_j, where $j = \arg\max_j sim(d, \overrightarrow{c_j})$. The centroid of the cluster C_j is modified accordingly. Otherwise, if *sim* < *thres*, then the story *d* forms a new cluster. The whole clustering procedure starts with an empty list of clusters and the first story on the input forms the first cluster.

Obviously, such a clustering algorithm may be used both in the off-line mode when the whole corpus of stories is available for clustering right from the beginning, as well as in the on-line mode when the stories arrive one by one. In the latter case, this algorithm solves also the problem of the first story detection problem for which some other approaches will be discussed later. A weak point of this algorithm is the need to set the value of the threshold *thres*. This parameter can be tuned experimentally but this may not always work due to the very nature of the problem which assumes that new topics, unknown in advance, are expected to appear all the time during the processing of stories.

In general, the *first story detection* may be seen as a task of the binary classification which requires distinguishing the incoming stories as starting a new topic or not. Often, this binary decision is based on a score computed by the classifier and compared with a threshold which may be tuned in the training process. A simple solution consists in comparing a given story with the window of *k* previous stories and checking its distance to the most similar from among them, possibly taking into account the time difference (i.e., older stories in the window are treated as less similar to a given story, by definition). If this distance exceeds a predefined threshold value, then the story is judged as starting a new topic.

Another idea consists in the monitoring of terms distribution over time and a new topic is recognized in case a sudden change in this distribution is detected for a given story. Still another option is to use the incremental clustering algorithm, as discussed in the previous passage. It may be modified in the following way so as to pay more attention to the recent stories. A limit N_C is imposed on the number of clusters preserved by the algorithm. Now, if a new cluster is formed when an incoming story is recognized as a first story, and there already exist N_C clusters, then the oldest cluster is dropped.

The *topic tracking* task is basically very similar to the text categorization problem or, its special case, the so-called *document routing* or *filtering* (Baeza-Yates and Ribeiro-Neto, 2011). Thus, the topic tracking problem may be solved using similar methods. For example, a query may be derived from the set of stories known to belong to the topic under consideration (positive examples) as well as from the stories known not to belong to it (negative examples). Then, such a query is executed against each incoming story and if there is a match, the story is recognized as belonging to the topic. Other popular classification algorithms such as the *k*-NN and decision trees have also been used.

3.3 Evaluation measures

The evaluation of solutions proposed to solve the TDT task is based on the classic measures, e.g., for the first story detection the classic confusion matrix based measures of the classification effectiveness are employed while for the retrospective topic detection these are the measures used to evaluate the clustering algorithms in case the ground-truth, i.e., actual partition of the stories set, is known. For convenience, the evaluation is carried out for each topic separately, i.e., the training and testing datasets are formed for each topic and the proposed algorithms are evaluated on them independently. It has to be emphasized that the notion of the training dataset is here rather specific as, e.g., in the case of the first story detection task or in the topic tracking task the algorithms are not necessarily trained on these datasets but rather they are just a collection of the background/contrast data used when the stories from a test dataset are classified.

The topic tracking is evaluated assuming that some set of stories $D_c \subseteq D$ belonging to a topic c is given and incoming stories are to be decided as to belong to this topic or not. Some variants emerge depending on whether a story detected by the evaluated system as belonging to the topic is added to the set D_c or not. Thus, this is a problem of the binary classification and again the confusion matrix based effectiveness measures are applicable for its evaluation.

There were a few data collections prepared for the purposes of the TDT solutions evaluation. The first one (Allan, 1998) comprises 15683 stories from the CNN and Reuters with 25 topics distinguished and manually assigned to the stories. The next one has already gathered 57000 stories with 100 topics distinguished.

4. A novel TDT related text classification problem

4.1 Problem statement

Recently, we have formulated a new text categorization problem that has been strongly inspired by some relevant practical problems. The original inspiration for this problem is the way the documents have to be handled by public institutions in Poland. Namely, such institutions are required by law to organize their documents, both incoming and produced themselves, into *cases* which in turn belong to some *topics*. Cases may be seen as sequences of documents which concern a specific matter and have been produced as a result of an instance of a business process carried out by given institution. An example of such a process, and a related case, may be a meeting of the advisory council. The first document in such a case may be a decision to organize such a meeting signed by some authorized person, e.g., a chairman of a council or board. A next document may be an official announcement which is distributed among relevant persons and so on, and the last document may be for example minutes of a protocol of the meeting. All these documents have to be organized in one case and this case have to be assigned to a proper topic. The eligible topics are precisely specified in the rules of conduct of the institution and form a hierarchy. In case of the previous example, a high level topic may be named “Advisory Council activities” and its descendant in the hierarchy topic may be exemplified by “Advisory Council meetings”. Documents within a case are chronologically ordered according to the date a document has been created or received.

The related categorization problem, referred to as the CCC problem (Category/Case Classification problem) may be thus defined as follows. Let us denote the set of documents as D and a set of categories as C, as previously. Additionally, let us introduce the following notation for cases (sequences of documents) and their sets:

- $\sigma_k = <d_{k_1}, \cdots d_{k_l}>$ is a *sequence of documents* (*case*),
- $\Sigma=\{\sigma_1,..., \sigma_p\}$ is a set of cases; all documents of a case belong to the same category c.

Let us assume that there is a set of cases, Σ, present in the system. Now, a new document d arrives and has to be classified to a proper case within a proper category c_j. Such a proper case may be one of the cases already existing in the system, $\sigma_i \in \sum$, or a new case which has to be established in a proper topic. It is assumed that the classification of a document to a proper category follows the text categorization paradigm. Namely, documents belonging to the same category are thematically and possibly also structurally similar (e.g., protocols of a council meetings etc.). On the other hand, documents belonging to the same case, while also similar in the above sense, are additionally forming a logical sequence and correspond to the subsequent stages of an evolving business process.

Thus, the classification of documents to cases is more in the spirit of the TDT problem. There is a number of similar specific tasks which have to be addressed in case of both the CCC and TDT problems. However, there are still some aspects which are different or which have been so far not considered in the context of the TDT. These include the following.

In the CCC problem an explicit set, or even hierarchy, of categories is assumed and each case (topic) belongs to a category while such a set of categories is not part of the original TDT problem. However, the idea of the hierarchical topic detection and tracking (HTDT) (Allan et al., 2003) is fairly similar to that of the CCC. Thus, the set C of categories is in the CCC the same concept as in the text categorization problem (cf. (2)). On the other hand, cases in the CCC correspond to the topics in the TDT.

In both cases the stories are time stamped. This is usually exploited in the TDT only in such a way that, e.g., older documents are considered in the first story detection or topic tracking to a limited extent or are even totally ignored. However, in case of the TDT the evolution of the content of subsequent stories belonging to a given topic is practically not taken into account. This is due to the fact that in the TDT the stories are assumed to be incoming from different sources, possibly multilingual, and thus there may be many stories describing the same aspect and/or the same stage of evolvement of the event laying at the ground of a given topic, e.g., produced by different new agencies. In case of the CCC problem subsequent documents (stories) are produced due to the development of the underlying case (a business process) and thus it may be assumed that they describe subsequent stages of a case which makes it possible to develop algorithms, potentially more effective, exploiting this characteristic of the task. This does not mean that techniques exploiting such an evolution of the stories content within a topic are not conceivable for the TDT problem. By definition, a TDT topic comprises stories on a triggering event and related events, thus the evolution of the content may be observed also here but due to the mentioned possible lack of “linearity” of the content, it is more difficult to be exploited.

There are other subtle differences between both problems. For example, in case of the first story detection task (FSD), in the CCC we assume that the system actually sees whole cases, i.e., when the system is “turned on” the ongoing cases are properly represented by the sequences of documents and when a new case starts then the system will see its first document on its input. This makes it possible to look for some characteristics of first documents of cases belonging to particular categories what may help to solve the FSD task. On the other hand, for the TDT it is assumed that the system may be "turned on" somehow "in the middle" of some topics evolution and first stories concerning these topics which are visible to the system may be far from the actual first stories (Allan, 2002).

4.2 Proposed solutions

In (Zadrożny et al., 2013) we formally introduced the novel CCC problem and proposed two approaches to solve it. We will now briefly recall the idea of these approaches which basically belong to the realm of the supervised learning. The training data set comprises a collection of documents D, arranged in a number of cases, Σ. In the testing phase another similar data set is used but some of the cases are divided into two parts. The first part is preserved as an on-going case while the documents of the second part are used, preserving their order, as documents to be classified. The evaluation measure employed is the percentage of documents assigned to a proper case.

Both proposed approaches are focused on a direct assignment of a document to a case. Their underlying idea is to learn the logic governing the sequence of documents forming a case. It is assumed that this logic is different for different categories and has to be learnt separately.

The documents are represented as vectors over a space of features which may be keywords (terms) from a set $T, T = \{t_1, \ldots, t_m\}$, following the classical vector space model (Baeza-Yates and Ribeiro-Neto, 2011), or *topics* identified using the Latent Dirichlet Allocation modelling (Blei et al., 2003) or any other entities used in various approaches to the modelling of documents within the information retrieval realm which basically follow a similar philosophy.

The first approach employs the popular technique to the modelling of sequences, namely the Hidden Markov Model (HMM) (Rabiner, 1989). We want to model the succession of documents in a case that is specific for particular categories. The hidden states of an HMM may be identified with the stages of a business process underlying the category of cases under consideration. For example, the following stages may be recognized in the cases related to advisory council meetings: decision, announcement, list of attendance, minutes etc. Of course, such stages do not have to be identified explicitly as the hidden states are identified in the data driven process of an HMM learning. The order of the stages of a business process is in general not deterministic and some stages may be repeated several times – all these aspects are addressed by the probabilistic nature of the HMM.

Formally, an HMM is defined by specifying the following elements: *the number of hidden states L,* the set of the hidden states may be thus denoted as $S = \{S_1, S_2, \ldots, S_L\}$ and a state in a time moment t will be denoted as q_t; *observations* generated by an HMM in subsequent states, corresponding here to the representation of the whole documents d forming a case or to individual keywords/terms present in the representation of these documents; *a state transition matrix* $A = [a_{ij}]$ defining the probability of transition from one state to another, $a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$, $1 \leq i, j \leq L$, *observation probability distributions* b_j defined for each state j in an appropriate space; an *initial probability distribution* in the space of states, $\pi = [\pi_1, \pi_2, \ldots, \pi_L]$ where $\pi_j = P(q_1 = S_j)$, $1 \leq j \leq L$.

A separate HMM, denoted as λ_c, is assumed for each category $c \in C$, and is trained using a collection of complete cases belonging to this category. The number of states L is set for each HMM based on some experimentation and taking possibly into account the average length of cases belonging to a given category.

A new incoming document d to be classified is matched against each case present in the system, $\sigma = \langle d_1, d_2, \ldots, d_p \rangle$. For this purpose, a matching degree $md(\sigma,d)$ of the document d against the case σ is computed as the conditional probability that the HMM λ_c, trained for the category c to which the case σ belongs, *will* generate the sequence of documents currently forming the case σ extended with the document d:

$$md(\sigma,d) = \mathrm{P}(d_1, d_2, \ldots, d_p, d \mid d_1, d_2, \ldots, d_p, \lambda_c) = \frac{\mathrm{P}(d_1, d_2, \ldots, d_p, d \mid \lambda_c)}{\mathrm{P}(d_1, d_2, \ldots, d_p \mid \lambda_c)} \tag{5}$$

In order to cover also the situation that the document d should start a new case, an „empty" case, comprising no documents, is also assumed to be always present in each category.

Then (5) takes the following form:

$$\mathrm{P}_\sigma(d \mid \lambda_c) = \sum_{j=1}^{L} \pi_j b_j(d\ ) \tag{6}$$

The document d is assigned to such a case σ^* that:

$$\sigma^* = \arg\max_{\sigma} md(\sigma, d) \tag{7}$$

The second approach to solving the CCC problem employs the *sequence mining* (Agrawal and Srikant, 1995; Zaki*,* 2001) to reveal the logic behind the succession of documents in cases of a category. Let us denote the set of keywords/terms used to index/represent the documents as T. Documents $d_i \in D$ are represented as sets of keywords, $d_i \subseteq T$, and our aim is to characterize cases of a particular category with sequences of groups of keywords appearing frequently is subsequent documents. More formally, let $f_i \subseteq T$ be a set of keywords and let $F = < f_1, f_2, \ldots, f_r >$ be a sequence of sets of keywords. The sequence F will be said to appear in a case $\sigma = < d_1, d_2, \ldots, d_s >$ if there exists such a subsequence of documents of σ, denoted $\sigma_r = < d_{i_1}, d_{i_2}, \ldots, d_{i_r} >$, that $i_k < i_l$ and $f_j \subseteq d_{i_j}$ (a document is represented as a set of keywords, as mentioned earlier). A sequence of sets of keywords F is said to be *frequent* in a given set of cases Σ if it appears in the number of cases which exceeds a certain threshold value.

An algorithm such as SPADE (Zaki, 2001) makes it possible to find all frequent sequences of sets of keywords for a given set of cases. Based on the frequent sequences, it is possible to define *rules* which describe dependencies between the occurrence of particular sets of keywords. In particular the following rules are of interest:

$$\text{if } F =< f_1, f_2, \dots, f_r > \text{ is frequent then } G =< f_1, f_2, \dots, f_r, g_{r+1} > \text{ is frequent} \qquad (8)$$

Thus, the approach to the CCC problem based on sequence mining works as follows. The sets of cases, for each category separately, are mined and the rules such as (8) are derived. A new incoming document d is matched against each case σ in the following way. Each rule (8) derived for a category *c* to which the case σ belongs and the left hand side *F* of which appears in σ is considered. Among them the rules with g_{r+1} (cf. (8)) being a subset of the document *d* are counted. The document d is classified to the case for which the number of rules counted in the previous step is the highest provided that this number is higher than a certain predefined threshold value. If this threshold value is not exceeded for any case then the document d starts a new case in the category which is selected using a standard text categorization algorithm.

For a more detailed description of both approaches and some experimental results the reader is referred to (Zadrożny et al., 2013).

5. Conclusions

We have discussed a new text classification problem which shares its main characteristic features with both the classical text categorization problem and the topic tracking and detection problem. We have reminded the essence of these two latter already classic areas of information retrieval and confronted with them our new problem formulation. Its main original aspect is the focus on the modelling of the succession of documents grouped into what is known as a topic in the TDT. The future research concentrates on developing new more efficient and effective algorithms than those considered so far.

6. Acknowledgment

This work is supported by the National Science Centre (contract no. UMO-2011/01/B/ST6/06908).

References

Agrawal R., Srikant R. (1995) Mining Sequential Patterns. *Proceedings of the 11th International Conference on Data Engineering (ICDE), Taipei, Taiwan, March 1995*, 3-14.

Allan J. (2002) Introduction to Topic Detection and Tracking. Event-based Information Organization. In: Allan J. (ed.): *Topic Detection and Tracking*, Springer, 1-16.

Allan J., Carbonell J., Doddington G., Yamron J.P., Yang Y. (1998) Topic Detection and Tracking Pilot Study: Final Report. *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, February 8-11, 1998 (http://www.itl.nist.gov/iad/mig//publications/proceedings/darpa98/pdf/tdt2040.pdf)

Allan J., Feng A., Bolivar A. (2003) Flexible intrinsic evaluation of hierarchical clustering for TDT. *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management (CIKM 2003)*, ACM, 263-270.

Baeza-Yates R.A., Ribeiro-Neto B.A. (2011) Modern Information Retrieval - the concepts and technology behind search, Second edition. Pearson Education Ltd., Harlow, England 2011.

Blei D.M., Ng A. Y., Jordan M. I. (2003) Latent Dirichlet Allocation. *Journal of Machine Learning Research,* 3, 993-1022.

Everitt B.S., Landau S., Leese M., Stahl D. (2010) Cluster Analysis, 5th Edition, Wiley.

Hayes Ph. J., Weinstein S. P. (1990) CONSTRUE/TIS: A System for Content-Based Indexing of a Database of News Stories. In: Rappaport A. T. and Smith R. D., Eds., *Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence (IAAI '90),* AAAI Press, 49-64.

Kumaran G., Allan J. (2005) Using names and topics for new event detection. In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 121-128.

Letsche T.A., Berry M.W. (1997) Large-Scale Information Retrieval with Latent Semantic Indexing. *Information Sciences*. **100,** 1-4, 105-137.

NIST (2008) National Institute of Standards and Technology (NIST) website devoted to the TDT. http://www.itl.nist.gov/iad/mig//tests/tdt/

Rabiner L. (1989) A tutorial on HMM and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 2, 257-286.

Salton G., ed. (1971). The SMART Retrieval System. Englewood Cliffs, N.J. Prentice Hall.

Sebastiani F. (1999) A tutorial on automated text categorization. In: A. Amandi, A. Zunino (Eds.), *Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, Buenos Aires, Argentina, 1999*, 7-35.

Sebastiani F. (2002) Machine learning in automated text categorization. *ACM Computing Surveys*, **34**, 1, 1-47.

Sebastiani F. (2005) Text Categorization. In: L.C. Rivero, J.H. Doorn, V.E. Ferraggine (Eds.) *Encyclopedia of Database Technologies and Applications*, Idea Group, 683-687.

Wayne C.L. (2000) Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation. In: *Proceedings of the Second International Conference on Language Resources and Evaluation, LREC 2000, 31 May - June 2, 2000, Athens, Greece. (LREC 2000)*.

Yang Y., Liu X. (1999) A re-examination of text categorization methods. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99)*. ACM, 42-49.

Zadrożny S., Kacprzyk J., Gajewski M., Wysocki M. (2013) A novel text classification problem and its solution. *Technical Transaction. Automatic Control*, **4-AC**, 7–16.

Zaki M.J. (2001) SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*, 42, 1-2 (January 2001), 31-60.