

NewsExplorer – combining various text analysis tools to allow multilingual news linking and exploration

Ralf Steinberger & Bruno Pouliquen

The screenshot displays the NewsExplorer web application interface. At the top, there are navigation tabs for 'EMM NewsBrief', 'EMM NewsExplorer', 'Name Search', and 'Text Search'. The main header includes the 'EMM NewsExplorer' logo and the text 'News Analysis RSS feed for the latest news summary Daily News Analysis, across languages and over time'.

The interface is divided into several sections:

- Menú principal:** Includes 'News Summary' and 'About EMM NewsExplorer'.
- News language and date:** A dropdown menu for 'Language or country:' is set to 'es - Español'. A date selector shows 'Jun 2008' with a calendar grid highlighting the 9th.
- Clustered news for lunes 9 de junio de 2008:** A world map shows news clusters. Below it, a list of countries and their article counts is provided:

Countries	Count
España	585
Estados Unidos	494
Francia	170
Argentina	108
Venezuela	88
Reino Unido	79
Perú	79
Ecuador	73
Japón	73
Alemania	71
Bolivia	65
China	55
Irak	52
Austria	50
Chile	48
Argelia	47
Ucrania	44
Colombia	43
Grecia	41
Irán	39
Brasil	38
Rusia	36
Suiza	27
Portugal	27
Italia	24
México	24
- This Week's New Stories:** A list of recent news items with dates and brief descriptions.
- This Month's New Stories:** A list of news items from the current month.
- Analysis over time:** A timeline and bar chart showing news volume over time.
- Pervez Musharraf:** A detailed profile section including:
 - Names:** A list of names in various languages (e.g., Pervez Musharraf (Eu,sv), General Pervez Musharraf (da,sv), Gen Musharraf (en), etc.).
 - Key Titles and Phrases:** A list of titles and phrases in various languages (e.g., pakistani president (en - 827), president (de,sv - 3230), etc.).
 - External resources:** A link to a Wikipedia image of Pervez Musharraf.
- Castro quits as president, state-run paper reports [72]:** A news snippet with a list of language links: 'de es fr it nl ar bg da et fa no pl pt ro ru sl sv tr'.
- Fidel Castro announced his resignation as president of Cuba and commander-in-chief of Cuba's military on Tuesday, according to a letter published by state-run newspaper Granma.** A news snippet with a list of language links: 'de en es fr it nl ar bg da et fa no pl pt ro ru sl sv tr'.
- Network Graph:** A complex network graph showing connections between various news items and entities, with nodes and edges representing relationships.

NewsExplorer – combining various text analysis tools to allow multilingual news linking and exploration

Ralf Steinberger & Bruno Pouliquen

and the present and past *Web Mining and Intelligence* Team

European Commission – Joint Research Centre

21027 Ispra (VA), Italy

Contact: Ralf.Steinberger@jrc.it

Online applications: <http://press.jrc.it/overview.html>

Technical Information: <http://langtech.jrc.it/>

Abstract. NewsExplorer (<http://press.jrc.it/NewsExplorer>) is a freely accessible, multilingual online application for news aggregation, analysis and exploration. It processes an average of about 30,000 news articles per day, gathered from about 1,400 news portals on the web. For each of the 19 languages covered, it groups related articles every day into clusters, extracts names of persons, organisations and locations from these clusters, links the clusters across languages, and aggregates historically related clusters into longer so-called stories. For the entity types person and organisation, it gathers and aggregates extracted information from all languages and over time. The results for each entity are displayed on dedicated web pages. For each entity, users will thus find: lists of latest news clusters and stories where the entity was mentioned, lists of other entities found in the same clusters, titles and other phrases describing the entity, quotations by and about this entity, and a photograph and a link to the corresponding Wikipedia site, when available. NewsExplorer makes use of – and has integrated fully – a number of different text mining techniques including clustering, multi-label categorisation, keyword extraction, named entity recognition and disambiguation, quotation recognition, script transliteration, name variant matching, topic detection and tracking, as well as cross-lingual document similarity calculation. The most outstanding features of NewsExplorer are its high multilinguality (currently 19 languages) and especially its capability to link and aggregate information across all languages and language pairs. The lecture will present NewsExplorer and – briefly – the other JRC-developed online news aggregation applications (see <http://press.jrc.it/overview.html>). It will then describe each of the components in some detail. The presentation will duly highlight the specific features and design decisions that allowed to achieve the high multilinguality of the application

Keywords. NewsExplorer; news analysis; news aggregation; multilinguality; clustering; classification; named entity recognition; social networks; name variant matching; quote detection; geo-tagging; topic detection and tracking; cross-lingual document similarity calculation..

1 Introduction

This text is the documentation accompanying the lecture with the same name at the Summer School *Curso de Tecnologías Lingüísticas*, held at the *Fundación Duques de Soria* in Soria, Spain on 10 July 2008. It gives slightly more detail and background information than will be presented during the two-hour talk. Its purpose is to prepare the interested students thoroughly for the lecture. As Sections 3 to 9 each describe different text analysis components, it is possible to skip one or more of them while reading this document.

1.1 Purpose of this document

The purpose of the lecture and of this documentation is to *give an overview of the NewsExplorer application and of its various text analysis components*. NewsExplorer (<http://press.jrc.it/NewsExplorer/>) is a multilingual news gathering, aggregation, analysis and exploration system developed at the European Commission's (EC) *Joint Research Centre* (JRC) in Ispra, Italy. NewsExplorer is part of the *Europe Media Monitor* (EMM) family of applications (see <http://emm.jrc.it/overview.html>), which furthermore

include *NewsBrief* (live news aggregation, clustering and classification), the *Medical Information System MedISys* (news aggregation, filtering, classification, trend detection and alerting) and *EMM-Labs* (a collection of various other recent text analysis and visualisation tools, including: the extraction of violent events and natural disasters; a social network browser; automatically generated monthly country reports, and more).

1.2 NewsExplorer functionality

The following is a list of NewsExplorer's functionality:

- Via the EMM core engine (Best et al. 2005), continuous collection of the most recent news items in currently 42 languages from about 1500 web portals world-wide; extraction of the text part of web pages and transformation to UTF-8-encoded RSS format.
- Separately for each of the 19 languages covered by NewsExplorer, clustering of the news of each day (24-hour period) and presentation according to cluster size; for each cluster, identification of the most typical article (the medoid article) and its title. The languages currently covered by NewsExplorer are: Arabic, Bulgarian, Danish, Dutch, English, Estonian, Farsi, French, German, Italian, Norwegian, Polish, Portuguese, Romanian, Russian, Slovene, Spanish, Swedish and Turkish.
- For ten countries and regions, NewsExplorer allows to display clusters consisting only of articles published in that country or region. This is relevant for languages that are spoken in more than one country. For instance, users may want to see only news published in their own country (e.g. the UK, the US, Germany or Austria). This functionality is also relevant for multilingual countries or regions, as users may want to see all news published in their country or region, independently of the language (e.g. French and Flemish in Belgium; English or French or Portuguese, etc. on the African continent).
- Identification and display – separately for each cluster – of references to persons, organisations and geographical locations.
- Linking of news clusters to related news clusters identified during the previous 7 days, and aggregation of such historically related news clusters into *stories*, which can last up to years if media reporting is continuous. A *story* within NewsExplorer is thus defined as a collection of news clusters linked over several days.
- Linking of news clusters to related news clusters across languages.
- For each person and organisation (referred to as *entities*) identified during the news analysis process, storage of meta-information in a relational database and display of the information on dedicated web pages. The extracted meta-information includes:
 - a. Latest clusters and stories mentioning this entity.
 - b. Name variants found for this entity
 - c. Titles and other accompanying phrases found for this entity
 - d. Quotations by and about this entity.
 - e. Persons frequently mentioned in the same news clusters (referred to as *related people*).
 - f. Other entities (mostly organisations and events) mentioned in the same news clusters (referred to as *other names*).
 - g. Persons mentioned particularly together with this entity and not so much with other entities (referred to as *associated people*).
- For each story, meta-information collected since the story started is shown, including:
 - a. The names of the first and of the biggest clusters of this story.
 - b. Most relevant country names, entity names and keywords.
 - c. An interactive timeline showing the number of articles per day pertaining to this story.
 - d. Names of, and links to, the latest clusters belonging to this story.
 - e. Further meta-information such as start and end date of the story, and the number of articles and clusters belonging to this story.
 - f. Related people, associated people and other names specifically for this story.

1.3 Text analysis components used in NewsExplorer

NewsExplorer consists of a whole range of text analysis components. These include:

- Monolingual document clustering (clustering).
- Recognition and disambiguation of the named entity types person, organisation (or – in a few cases – events) and geographical location (named entity recognition and classification; information extraction).

- Recognition of quotations and reference resolution for name parts.
- Identification and mapping of name variants for the same person, both within the same language and across languages and writing systems (e.g. Cyrillic and Arabic) (transliteration, string similarity calculation; name variant mapping).
- Categorisation of documents according to the multilingual thesaurus Eurovoc (multi-label document categorisation).
- Cluster similarity calculation, both monolingual and across languages (topic tracking and detection; cross-lingual document similarity calculation).

Every day, and separately for each of the 19 languages covered, NewsExplorer groups related articles into *news clusters*. Clusters are computed using a group average agglomerative bottom-up clustering algorithm (similar to Schultz & Liberman 1999). Each article is represented as a vector of keywords with the keywords being the words of the text (except stop words) and their weight being the log-likelihood value computed using word frequency lists based on several years of news. We additionally enrich the features with the countries mentioned in the article (Pouliquen et al. 2004).

Each computed cluster consists of its keywords (i.e. the average log-likelihood weights for each word found in the individual texts) and the title of the cluster's medoid (i.e. the article closest to the centroid of the cluster). In addition we enrich the cluster with features that will be used in further processes (Pouliquen et al. 2004):

- the date, language and weight (number of articles in the clusters);
- lists of countries mentioned in the cluster, produced with a geo-tagger that disambiguates place names at the cluster level (see Section 3);
- lists of person and organisation names extracted from the articles of the cluster (see Sections 4 and 5);
- a vector of subject domain descriptors from the multilingual Eurovoc thesaurus (see Section 7): this is a more 'semantic' index of descriptors like *Natural Disaster*, *Rural Migration* or *Fishing Quota*.

Section 8 will show that the lists of keywords and the country information are useful ingredients to link related news clusters over time (monolingual topic tracking), but that the whole range of ingredients are required for cross-lingual cluster similarity calculation.

1.4 Key publications on NewsExplorer

Due to the variety of applications to be discussed in this document and the limited time available during the session, references to related work and state-of-the-art discussions will be relatively short and sketchy. For all of the applications described here, however, there are previous publications where readers can find more detail regarding the state-of-the-art, the method used, and evaluation results. The major previous publications describing individual NewsExplorer tools in detail (partially reused when rewriting this text) are listed below. At http://langtech.jrc.it/JRC_Publications.html, many of the publications are available for download:

- Steinberger Ralf, Bruno Pouliquen & Camelia Ignat (forthcoming). *Using language-independent rules to achieve high multilinguality in text mining*. In: Françoise Fogelman-Soulié, Domenico Perrotta, Jakub Piskorski & Ralf Steinberger (eds): *Mining Massive Data Sets for Security*. IOS-Press, Amsterdam, Holland. (Discussion of the design decisions taken to be able to deal with the currently 19 NewsExplorer languages).
- Pouliquen Bruno & Ralf Steinberger (forthcoming). *Automatic Construction of Multilingual Name Dictionaries*. In: Cyril Goutte, Nicola Cancedda, Marc Dymetman & George Foster (eds.): *Learning Machine Translation*. MIT Press, NIPS series. (multilingual named entity recognition, disambiguation and name variant mapping).
- Pouliquen Bruno, Ralf Steinberger & Camelia Ignat (2003). *Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus*. In: Proceedings of the EUROLAN Workshop Ontologies and Information Extraction at the Summer School The Semantic Web and Language Technology - Its Potential and Practicalities. Bucharest, Romania. (multi-label document categorisation using the Eurovoc thesaurus).
- Pouliquen Bruno, Marco Kimler, Ralf Steinberger, Camelia Ignat, Tamara Oellinger, Ken Blackler, Flavio Fuat, Wajdi Zaghouni, Anna Widiger, Ann-Charlotte Forslund, Clive Best (2006). *Geocoding multilingual texts: Recognition, Disambiguation and Visualisation*. Proceedings of the 5th International Conference on Language Resources and Evaluation LREC, pp. 53-58. Genoa, Italy. (geo-tagging, i.e. recognition and disambiguation of geographical references in texts).

- Pouliquen Bruno, Ralf Steinberger & Clive Best (2007). *Automatic Detection of Quotations in Multilingual News*. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP. Borovets, Bulgaria. (multilingual recognition of quotations).
- Pouliquen Bruno, Olivier Deguernel & Ralf Steinberger (2008). *Story tracking: linking similar news over time and across languages*. In: Proceedings of the CoLing'2008 workshop: Multi-source, multilingual information extraction and summarization, Manchester, August 2008. (cluster similarity calculation; cross-lingual document similarity calculation)

1.5 The multilingual news data processed by NewsExplorer

Since 2002, the *Europe Media Monitor* (EMM) engine (see Best et al. 2005) gathers thousands of newspaper articles from many different sources around the world. Due to the European interests, news sources from within Europe are better represented than those from outside the European Union (EU). Currently (as of Spring 2008), EMM gathers an average of 50,000 news articles per day in 42 languages from about 1,500 news sources. EMM extracts the full article and its title from the original web pages, normalises them to UTF8-encoded RSS format, detects duplicate articles, keeps track of meta-information such as the source URL, the name of the news source and the time of download. EMM full-text indexes the articles and categorises them according to mostly customer-defined Boolean expressions.

NewsExplorer and its sister applications display the title, a link to the source URL and the first few words of each article plus the information extracted by its text analysis tools. If users want to read the whole news item, they are always redirected to the source URL where the original article was found.

1.6 Contents of the remainder of this document

The remainder of this document is organised as follows: Section 2 describes the general design principles which allowed NewsExplorer to be developed in more languages than most other existing text analysis systems. The next sections are dedicated to one specific application each: Section 3 describes the method used for geo-tagging. Section 4 is dedicated to the recognition and disambiguation of person and organisation names. Section 5 shows how NewsExplorer deals with inflection and other regular morphological variations. Section 6 focuses on how to automatically identify whether two different names are variants of each other (monolingual and multilingual variants of the same name; transliteration; how to deal with highly inflecting languages, etc.). Section 7 summarises the work to produce a multilingual subject domain signature for news clusters, using Eurovoc. This signature is used for cross-lingual cluster linking. Section 8 describes how related news are automatically identified over time and across languages. Section 9 is dedicated to the recognition of quotations (direct speech) including speaker assignment and identification on who the quotation is about. Section 10 briefly summarises this document. There is not a specific section on related work. Instead, references to related work will be given separately for each application, inside the sub-sections.

2 Achieving high multilinguality

A major feature of EMM and NewsExplorer are their multilinguality. Many alternative systems and text analysis applications are available for one language or for a small number of languages. Section 2.1 gives some practical advantages of systems that cover more languages. Section 2.2 explains how even larger numbers of languages can be covered if software development follows certain criteria.

2.1 Practical arguments for multilinguality

The text analysis functionalities discussed in this document are mostly available for the nineteen NewsExplorer languages. While many of these functionalities are applicable to information extracted from texts in a single language, the usefulness rises significantly if the information is derived from documents in many different languages. In NewsExplorer, about a quarter of the news articles are written in English as it is an international language and English news sources can be found for many countries of the world. However, there is ample evidence that much of the information would not be found if only English texts were analysed. Some examples are:

- Some areas of the world are better covered by different languages. For instance, North Africa is covered better by French, Brazil by Portuguese, and the rest of Latin America by Spanish. By analysing texts in more languages, more information will be found.
- Daily and long-term social network analysis has shown (Pouliquen et al. 2007b, Tanev 2007) that the dominant and most frequently mentioned personalities in English language news are the US President and the British Prime Minister. When looking at French, German, Arabic or Russian news, the

respective leaders have a much more central role. Adding analysis results from more languages reduces bias and increases transparency.

- NewsExplorer automatically collects name attributes (titles, professions, age, role, nationality, family relations, etc.) about the approximately 700,000 persons found in media reports in the course of several years from the news in different languages. The various attribute lists show clearly that the information found across languages is often complementary. To give a concrete example: For the murdered Russian spy Alexander Litvinenko, texts in most languages contained the information that Litvinenko was a Russian and that he was an agent. However, we found in French news that he was 43 years of age, in Italian news that he was killed, in German news that he was a *critic* of some sort, etc.¹

In a national context with a dominant language, for instance English in the USA, a possible solution to this multilinguality requirement is to use machine translation (MT) to translate the documents written in the most important foreign languages into the national language, and to then apply the text analysis tools to the texts in that language. In the European context, where there is not one, but 23 official languages, using MT is not an option, as there are 253 language pairs ($N * (N - 1) / 2$, with N being the number of languages involved). Multilingual text processing is generally important, but it is considerably more relevant in the European context than in most other settings. Another important issue is the fact that MT often fails when being confronted with proper names or specialist terminology (Pouliquen & Steinberger, forthcoming). Larkey et al.'s *native language hypothesis* (Larkey et al. 2004) – the observation that text analysis results are better if performed on the source language text – is also a good argument for applying multilingual text analysis rather than operating on machine-translated texts.

2.2 Design principles making it easier to achieve high multilinguality

There is more than one way to achieve multilinguality (Steinberger et al., forthcoming discuss this and list a number of other systems that cover several languages), but only few developers have explicitly mentioned the principles behind their development. The guidelines underlying the development of the NewsExplorer application are the following:

- Use language-independent rules wherever possible, i.e. use rules that can be applied to any new language that will be added to the system.
- Keep language-specific resources and tools to a minimum: Such tools are, for instance, linguistic software like part-of-speech taggers and parsers. Language-specific resources are morphological or subject domain-specific dictionaries, linguistic grammar rules, etc. As acquiring or developing such language-specific resources is difficult and expensive, it is better not to use them, or – where they are necessary – to use them as little as possible.²
- Keep the application modular by storing necessary language-specific resources outside the rules, in language-specific parameter files. That way, new languages can be *plugged in* more easily.
- For the language-specific information that cannot be done without, use bottom-up, data-driven bootstrapping methods to create the monolingual resources.
- Avoid language pair-specific resources and procedures because the almost exponential growth of language pairs would automatically limit the number of languages a system can deal with.

These are very simple and generic guidelines, but sticking to them is not always easy and sometimes comes at a cost. These principles are discussed in detail in Steinberger et al. (forthcoming).

3 Recognition and disambiguation of geographical references

The text mining task of *geo-tagging* (also referred to as *geo-coding* or as *grounding of geographical references*) consists of recognising references to geographical locations in free text and to identify unambiguously their (ranges of) latitude and longitude. Geo-tagging goes beyond the more commonly pursued task of *geo-parsing*, which consists of recognising the words or phrases without attempting to put a dot on a map. The latter can theoretically be done simply by looking at linguistic patterns, even if there is evidence that geo-parsing is difficult without a gazetteer (Mikheev et al. 1999). For instance, when we find the piece of text "... is located near XYZ", we can infer that XYZ is a place name of some sort. Geo-

¹ See <http://press.jrc.it/NewsExplorer/entities/en/97338.html> for more details regarding this example.

² Part-of-speech tagging software is freely available and can, in principle, be trained for any language, but we feel that the effort involved for training 20 languages is too big for the expected benefit. Using readily trained taggers is not an option for us, as the overheads of dealing with different tag sets and levels of performance, and with differing input and output formats, etc. are too big.

tagging, on the other hand, is not possible by looking at the text only. Additionally, a gazetteer, i.e. a list of existing places and their latitude-longitude and probably more information, is needed. A number of gazetteers are freely or commercially available, such as *GeoNames*³ and the *Global Discovery*⁴ database. These gazetteers usually contain place names in one or more languages, latitude and longitude information, size categories for each place (distinguishing capitals from major or minor cities, villages, etc.), as well as hierarchical information indicating that a town belongs to a county, which is part of a region, which is itself part of a country, etc. A third gazetteer, the KNAB⁵ database, is particularly useful as it contains a large number of exonyms (foreign language equivalents like *Venice*, *Venise* and *Venedig* for the Italian city of *Venezia*), as well as historical variants (e.g. *Constantinople* for the Turkish city of *Istanbul*).

Geo-tagging thus consists of finding gazetteer entries (and probably other expressions) in text. This is basically a lookup task, but there are a number of challenges that make this task much harder than it seems at first sight. We will present these briefly in the next section and will then present a language-independent rule set to deal with these issues. For a definition of the tasks and an overview of existing commercial and academic approaches, see the recent Ph.D. thesis by Leidner (Leidner 2007). For a more detailed description of the challenges, the proposed knowledge-poor solution and issues concerning the compilation of a multilingual gazetteer, see Pouliquen et al. (2006).

3.1 Challenges for Geo-tagging

When using a gazetteer, the first step in recognising place names in text is a look-up task. In all European languages, it is possible to compare only uppercase words in the text with the entries of the gazetteer. In languages not distinguishing case, such as Arabic or Chinese, every single word must be compared to the gazetteer. However, even for European languages, geo-tagging is more than a simple lookup task, as a number of ambiguities need to be solved and language-specific issues need to be tackled. The challenges are the following:

- (a) Homography between places and persons: For instance, there are both places and persons with the names of *George* (South Africa plus several other places world-wide with this name), *Washington* (capital of the USA and almost 50 other locations), *Paris* (French capital and almost 50 other locations) and *Hilton* (United Kingdom and many more places with this name).
- (b) Homography between different places: An example is *Alexandria*, as there are 24 different cities with this name in ten different countries: Greece, Romania, South Africa, USA, etc.
- (c) Homography between places and common words: For example, the English adjective *Nice* is also a city in France, the English verb *Split* is homographic with a major city in Croatia, etc. We have found a large number of places homographic with common words for all languages we worked with.
- (d) The same place has different names: This is not only true across languages (e.g. the Italian city of *Milano* has the ‘translations’ – or exonyms – *Milan*, *Milán*, *Mailand*, *Μιλάνο*, *Милан*, *Милано*, *ميلانو* etc.). Even within the same country – and sometimes within the same language – places can have several names. Examples are *Bombay/Mumbai* and *Bruxelles/Brussell*.
- (e) Place names are declined or morphologically altered by other types of inflection. This has the consequence that the canonical form found in the gazetteer will frequently not coincide with the declension found in the text. The US-American city of *New York* can for instance be referred to as *New Yorkului* in Romanian or as *New Yorgile* in Estonian.

3.2 Language-independent rules for Geo-tagging

Challenges (d) and (e) necessarily involve the usage of language-specific resources: If name variants like *Mailand*, *Milán* or *Милан* are not in the gazetteer (challenge (d)), they cannot be looked up. The only solution to this problem is to find ways of populating an existing gazetteer with multilingual place name variants in the least time-consuming way. At the JRC, we have merged various gazetteers and additionally exploit the online encyclopaedia Wikipedia for this purpose. Wikipedia can even be used to find inflected forms of city names (challenge (e)). For instance, for the same city, the following inflection forms were found on the Finnish Wikipedia page: *Milanon*, *Milanossa*, *Milanosta*, *Milanolainen*, *Milanoon*, *Milanolaiset*, *Milanoa*. The temptation is big to simply use generous wild cards such as *Milana**, but due to the hundreds of thousands, or even millions of entries in a gazetteer and the likely similarity with other words of the language, this will lead to many wrongly recognised names, lowering Precision.

³ See <http://www.geonames.org/> (last visited 30.01.2008)

⁴ See <http://www.europa-tech.com/> (last visited 1.2.2008)

⁵ See <http://www.eki.ee/knab/knab.htm> (last visited 1.2.2008)

This specific wild card pattern would, for instance, wrongly recognise the Italian city of *Milano Marittima* and the Polish city *Milanówek*. The issue of inflection and the challenge to recognise other name variants will be discussed in Section 5, together with the variations of person names.

While challenges (d) and (e) cannot be overcome without language-specific resources and procedures, there are generic solutions for challenges (a), (b) and (c) that require no – or extremely little – language-specific effort. The proposed heuristics are the following:

- (a) For known names, prefer person name readings over location name readings: The idea is to ignore all potential locations that are homographic with a name part of a person talked about in the same document. For instance, if *Javier Solana* has been mentioned in the text, we should assume that any further reference to either *Javier* or *Solana* refers to the person and not to the respective locations in Spain and the Philippines. The reason for this rule is that person name recognition is more reliable than geo-tagging.
- (b) Make use of information about a place's importance or size: Many gazetteers like GeoNames or Global Discovery use size classes to indicate whether a place is a capital (size class 1), a major city, a town, etc. or a village (size class 6). If no further information is available from the context, we can assume that the text refers to the larger location. For instance, the Romanian city of *Roma* is of size class 4, while the Italian city with the same name is a capital (size class 1). The Italian capital should thus be chosen over the Romanian town.
- (c) Make use of the country context: the idea is that – if we already know that a text talks about a certain country – then it is likely that a homographic place name should be resolved in favour of the place in that country. For instance, if we know that the text talks about Romania because either the news source is from Romania or because another *non-ambiguous* reference is made to the country or any of its cities, the likelihood that we talk about the Romanian town *Roma* is much bigger.
- (d) Prefer locations that are physically closer to other, non-ambiguous locations mentioned in the text: In the case of ambiguity between two homographic places of the same size class, it is likely that the author meant to refer to the one nearby. For instance, there are two cities called *Brest*, one in France and one in Belarus. If both *Brest* and *Warsaw* are mentioned in a text, the latter reading will be preferred because it is at a distance of 200 km from Warsaw, while the French port is 2,000 km away.
- (e) Ignore places that are too difficult to disambiguate: Locations that are homographic with common words of a language frequently lead to wrong hits. Such locations should be put on a language-specific *geo-stop word list* and ignored if found in a text. If an author really intends to refer to the places called *And* (Iran) or *Had* (India, and others), these references will be missed, but many errors will be avoided. Such geo-stop word lists are language-dependent because the same words are likely not to be ambiguous in another language. For instance, *And* and *Had* are not ambiguous in German as there are no such words. Geo-stop word lists can be produced semi-automatically by comparing the gazetteer entries with a list of the most frequent words of a language and by hand-picking those words that were found. In order to reduce the work load, this list can be narrowed down to those words that appear more often in lowercase than in uppercase. The uppercase word *This* (potentially referring to a location in France), for instance, will be found in such a word frequency list, but the lowercase variant *this* will be much more frequent, meaning that *This* is a good candidate for the geo-stop word list. It is also advisable to compare the lexicon of a language with an international list of frequent first names.

These heuristics were derived from multilingual test data and were found to produce good results in a majority of cases (Pouliquen et al. 2006). The first four are completely independent of the text language as they refer to external parameters such as geographical location and location size. The fifth heuristic is language-dependent, but a good geo-stop word list for any given language can be produced within a few hours.

3.3 Combination of the rules

The rules mentioned in the previous section may contradict each other, so they have to be combined into a single rule that regulates their relative preference. When geo-tagging a new text, the binary rules will first be applied, i.e. geo-stop words will be ignored and potential locations that are homographic with part of a person name found in the text will not be considered. In a second instance, the formula below will be applied. The second formula explains the calculation of the parameter *kilometric weight*.

For computing the score, the current settings are:

$$\begin{aligned} \text{Score} &= \text{classScore} [80,30,20,10,5] \\ &+ 100 \text{ (if country in context)} \\ &+ 20 \cdot \text{kilometricWeight}() \end{aligned}$$

where: *classScore* is a given score depending on the importance of the place (this is coded by the *class* attribute: 80 for country name, capital or big city, 30 for province level city, 20 for small cities, 10 for villages, 5 for settlements); *kilometricWeight*, which has a value between 0 and 1, is the minimum distance between the place and all non-ambiguous places. This distance *d* is weighted using the arc-cotangent formula, as defined by Bronstein (1999), with an inflexion point set to 300 kilometres⁶, as shown in Equation 2.

$$\text{kilometric Weight}(d) = \frac{1}{\text{arcCot}\left(-\frac{300}{100}\right)} \text{arcCot}\left(\frac{d-300}{100}\right)$$

The formulae do not make reference to any language-specific features so that they can be applied to any new language without further consideration.

4 Named Entity Recognition and variant mapping

The names of *known* persons and organisations can be identified in new documents through a lookup procedure, just like the geographical place names from a gazetteer. For morphological and other variants, the same measures can be taken as for geographical names. These measures will be discussed in Section 5. However, as exhaustive lists of person and organisation names do not exist, *new names* need to be identified in one of two different ways: (1) When using dictionaries of a language, one could assume that any unknown word found in text is a proper name, but using dictionaries alone would be dangerous because any typo could then be identified as a name, as well. For languages that distinguish case and that begin proper names with an uppercase letter, the number of name candidates can of course be limited. (2) Additionally, or instead of using dictionaries, local patterns can be used (based on features like context words, part-of-speech information, syntactic parses, distance, or others), which can be either handwritten or acquired automatically with Machine Learning methods. In NewsExplorer, such local patterns are words, multi-word expressions or regular expressions that occur near to the names and that indicate that some (uppercase) words are likely to be a name. Such patterns can be titles (e.g. *Minister*), words indicating nationality (e.g. *German*), age (e.g. *32-year old*), occupation (e.g. *playboy*), a significant verbal phrase (e.g. *has declared*), and more. The words and expressions of different types can be referred to generically as *trigger words*, as their presence triggers the system to identify names. It goes without saying that these pattern recognition resources are necessarily language-specific. The challenge is thus to (a) keep the effort to produce these patterns to a minimum and (b) to formulate them in a generic way, which makes it easy to produce the patterns for a new language.

4.1 Patterns for name guessing

In JRC's named entity recognition tools, the generic patterns are basically language-independent, but they make use of language-specific resource files that contain the language-specific information such as lists of titles, nationality adjectives, etc. For full details on the used methods, see Pouliquen et al. (forthcoming) and Steinberger & Pouliquen (2007). Here, we will focus on the aspect of language independence and on how to create language-specific resources quickly and with little effort. One intrinsic feature of the JRC's tools is that they will only recognise names with at least two name parts. The reason for this is that the aim in NewsExplorer and other JRC tools is not only to recognise that *Angela* or *Bush* are names, but to identify the exact referent and its name variants so that users can search for all news items mentioning this person.

The following are the basic rules to identify person names in languages that write names with uppercase:

- (a) Any group of at least two uppercase words that is found to the left or to the right of one or more trigger words will be identified as a name candidate. Trigger words can also be regular expressions such as *[0-9]+-?year-old* to capture *43-year-old Alexander Litvinenko*.
- (b) The pattern allows the presence of a number of name infixes which can also be written in lower case, such as *von*, *van der*, *bin*, *al*, *de la*, etc. to also capture names such as *Mark van der Horst*, *Oscar de la Renta*, *Khaled bin Ahmed al-Khalifa*, etc.
- (c) Furthermore, a number of other frequent words are allowed between trigger words and the name. These can be determiners (e.g. *the*, *a*), adjectives and other modifiers (*former*, *wandering*), or com-

⁶ Empirical experiments showed that distances of less than 200 km are very significant, and distances more than 500 km do not make a big difference. Therefore, we have chosen the inflexion point between those two values: 300.

pound expressions (*most gifted*), allowing name recognition in sentences like “... *Juliette Binoche, the most gifted French actress*”.

- (d) Patterns should also allow for the inclusion of a slot for other names (e.g. *United Nations*), in order to capture expressions such as *Envoy to the United Nations*.
- (e) Names are also recognised if one of the potential name parts is a known first name. Large lists of first names from different languages and countries are thus a useful resource, that can be used for the name recognition in all languages. In the example *Angela UnknownWord*, the second element would thus be identified as the second name part. First names are thus different from the other trigger words mentioned earlier because they are part of the name, while titles and country adjectives are not.
- (f) Organisation names are different in that they are often longer and they can contain several lowercase words that are normal words of the language, as in Federal *Ministry of the Interior*, etc. In order to capture such names, the patterns must allow various sequences of typical organisation trigger words (e.g. *Bank, Organi[sz]ation, Ministry, Committee*, etc.), lowercase filler words (e.g. *of the*) and other content words (e.g. *Interior, Finance, Olympic*, etc.).

It is useful to also allow long sequences of trigger words to capture expressions like *former Lebanese Minister of Agriculture*. While the combination *Minister of Agriculture* may be enough to recognise the proper name, storing trigger words and their combinations has the advantage that they provide useful information on a person. In NewsExplorer, the historically collected trigger words are displayed together with each person.

4.2 Bootstrapping the acquisition of language-specific pattern ingredients

Besides the list of common first names, which can be used for the recognition of new names in any language, the various trigger words mentioned in the previous section are clearly different from one language to the other. These lists can be rather long. Our English list, for instance, consists of about 3400 trigger words. In order to compile such a list for a new language (e.g. Romanian), it is convenient to search a large corpus of that language for known names and to produce a frequency list of left and right-hand-side contexts of various sizes (e.g. between one and five words). The most frequent trigger words can then be manually selected. Bootstrapping will make the process more efficient: instead of going manually through the typically very long name context lists, it is better to use the most frequent trigger words found and to search the collection again for new names in order to collect more name contexts, and so on. Wikipedia or similar sources often contain lists of common first names and titles, so that such lists can also be used as a starting point.⁷ The effort to produce working lists of recognition patterns for a new language is between half a day and 5 person days. Note that the manual selection of trigger expressions is indispensable and that an intelligent reformulation of expressions can improve the rules massively. For instance, for Romanian occupations like *ministrul de interne*, experts can expand the rule immediately to other occupations: *ministrul de externe, ministrul de finanțe, ministrul justitiei, ministrul transporturilor* and more (Minister for external affairs, finance, justice and transport, respectively). They can even write a more complex pattern to allow the recognition of combinations like *ministrul transporturilor, constructiilor si turismului* (Minister of transport, construction and tourism) or *Ministrul delegat pentru comerț* (Vice-minister for commerce). In the case of Romanian, the first list has been validated to a certain extent and the expert produced 231 trigger words in about 3 hours of time. After this new list has been compiled, we launched the candidate extractor again and another validation was done by the expert. We now have 467 trigger words and Romanian is used fully in NewsExplorer, where it recognises an average of one hundred new names every day.

Another bootstrapping method would be to use MT or bilingual dictionaries in a triangulation approach, i.e. translating from two or more different languages into the new target language and to use only the overlapping results.

5 Dealing with inflection and other regular variations

For the tasks of geo-tagging, recognition of known person names and even for the guessing of new names in text, it is necessary to compare word forms in the text with lists of known words in lookup lists. Even though the task looks simple, looking up known entities in a new language is not always so straightforward because the word forms found in the text often differ from the word forms in the lookup tables.

⁷ See, for instance, the web site www.behindthename.com for first names, and the following page for lists of professions: <http://en.wikipedia.org/wiki/Category:Occupations> (available in various languages).

5.1 Reasons for the existence of name variants

The main reasons for these differences between the dictionary form in the lookup tables and the word forms found in real text – together with the adopted solutions – are the following:

- (a) Hyphen/space alternations: For hyphenated names such as *Jean-Marie*, *Nawaf al-Ahmad al-Jaber al-Sabah* or the place name *Saint-Jean*, we automatically generate patterns to capture both the hyphenated and the non-hyphenated forms (e.g. `Jean[\-\\]Marie`).
- (b) Diacritic variations: Words that carry diacritics are often found without the diacritic. For example, *François Chérèque* is often written *Francois Chereque*, Schröder as Schroder, Lech Wałęsa as Lech Walesa, Raphaël Ibañez as Raphael Ibanez, etc. For each name with diacritics, we therefore generate a pattern that allows both alternatives (e.g. `Fran(ç|c)ois Ch(é|e)r(ê|e)que`).
- (c) Further common variations: Especially for place names, there are a number of very common variations, including the abbreviation of name parts such as *Saint* to *St* (with or without the final dot) or the use of a slash instead of common name parts. For instance, *Nogent-sur-Seine* and *Frankfurt am Main* can be found as *Nogent/Seine* and *Frankfurt/Main* (also: *Frankfurt a. Main*), etc. For all such names, we thus pre-generate the various common variants.
- (d) Name inversion: While news texts in most languages mention the given name before the surname, the opposite order can also be found in some languages. In Hungarian, for example, local names are usually written with the last name first, while foreign names have the opposite order. The lookup procedure must consider this variation, as well.
- (e) Morphological declensions: In some languages (especially the Balto-Slavonic and Finno-Ugric languages), person names can be declined. In Polish, for example we can find the inflected form *Nicolasowi Sarkozy'emu* (or – less frequent – *Nicolasowi Sarkoziemiu*) referring to the French president *Nicolas Sarkozy*. Similarly *Tony'ego Blaira* or *Toniego Blaira* are found for the former British prime minister. For these languages, we pre-generate morphological variants for all known names according to rules that will be discussed below. It must be highlighted that – in some languages – variations can also affect the beginning of the name. For instance, for the Irish place name *Gaillimh* (Irish version of *Galway*), we can find *nGaillimh* (in Galway). For some languages, the number of different inflections can be rather high.
- (f) Typos: Even in the printed press, typos are relatively frequent. This is especially the case for difficult names such as *Condoleezza Rice*. For this person's given name, we found the typos *Condoleza*, *Condaleezza*, *Condollezza* and *Condeleeza*, each more than once.
- (g) Simplification: In order to avoid repetition, names such as *Condoleezza Rice* and *George W. Bush* are frequently simplified to *Ms. Rice* and *President Bush*.
- (h) Transliteration: Names are normally transliterated into the target language writing system. In the case of NewsExplorer, we are mainly interested in *Romanisation*, i.e. in using the Latin script as a target representation. Depending on the target language, transliteration rules often differ so that two different Romanised versions of the same name can co-exist. For example, the Russian name Владимир Устинов is typically transliterated to *Wladimir Ustinow* in German, to *Vladimir Ustinov* in English, to *Vladimir Ustinov* in Spanish, to *Vladimir Oestinov* in Dutch and to *Vladimir Oustinov* in French. Conversely the French city *Strasbourg* is sometimes written in Russian Страсбург (/strasburg/) sometimes Стразбург (/strazburg/), in Ukrainian Страсбур (/strasbur/), in Serbian Стразбур (/strazbur/), without the final 'g' mimicking the original French pronunciation.
- (i) Vowel variations, especially from and into Arabic: In Arabic and some other languages, short vowels are not always written. The string محمد (Mohammed) contains only the four consonants Mhmd, which is the reason why so many variants exist for this name: *Mohammed*, *Mohamed*, *Mahmoud*, *Muhamad*, and more.

5.2 Generating variants for known names

The variation types (a) to (d) are rather generic so that it is not difficult to pre-generate the most common name variants, as shown in the previous section. Morphological variations such as those shown in (e) are much harder to predict and they differ from one language to the next. This section describes how this variation type can be dealt with. For the variation types (f) to (i), see Section 6 below.

Profound linguistic skills and native speaker competence will help to produce good suffix addition and suffix replacement rules. Vitas et al. (2007), for instance, have produced extensive inflection rules for Serbian. However, in order to achieve high multilinguality, it is important to find an efficient and quick method to generate at least the most common variants. We found that, even without native speaker competence, it is possible to identify a number of frequent inflection rules purely by observing common varia-

tions for known names. These can be found by searching text collections using several generous regular expressions such as `Nicol.*Sarko[[:alpha:]]+` (for the French president) allowing to capture name variants and by then looking for regularities. In Romanian, for example, one will discover that known names are frequently accompanied by the suffixes `-ul` and `-ului` (suffix addition), and that the endings are `-l` and `-lui` if the name already ends in `-u`. If the name ends with `-a` we frequently find the `-ie` ending (suffix replacement). By collecting a number of such observations, we can produce suffix addition and replacement rules to pre-generate – for all names in the lookup tables – the most frequent variants. For the names *Paris*, *Bacău* and *Venezia*, for example, we can then generate the variants *Parisul*, *Parisului*, *Bacăul*, *Bacăului*, *Venezia* and *Veneziei*. Slavic languages are more complex, but the same principle holds. For the Slavic language Slovene, the regular expression substitution rule for person names is:

```
s/[aeo]?/(e|a|o|u|om|em|m|ju|jem|ja)?/
```

meaning that – for every name ending in `-a`, `-e` or `-o` – we pre-generate ten different variants, ending in `-e`, `-a`, `-o`, `-u`, `-om`, `-em`, `-m`, `-ju`, `-jem` and `-ja`, respectively. For every frequent known name in our name database such as the previous Lebanese political leader *Pierre Gemayel*, for instance, we will thus generate the pattern:

```
Pierr(e|a|o|u|om|em|m|ju|jem|ja)? Gemayel(e|a|o|u|om|em|m|ju|jem|ja)?
```

That way, *Pierrom Gemayelom* and any of the other possible combinations will be recognised in text. Note that over-generation, i.e. producing name variants that do not exist in the language, is not normally a problem because they will simply not be found. However, especially short names can lead to over-generous patterns and new patterns should always be tested on large document collections before being applied in a real-world scenario.

To give an indication of the effort required: it takes us usually between 1 hour and 2 days to produce basic working lists of inflection patterns for a new language.

An alternative to *generating* morphological variants for highly inflective languages would be to map different name variants found in text using a mixture of string distance metrics and automatically acquired suffix-based lemmatisation patterns, as it was proposed by Piskorski et al. (2008) for Polish.

5.3 Transcription of names written in different writing systems

Names from a language using a different writing system are usually transliterated. Transliteration rules are simple, normally hand-written rules mapping characters or sequences of characters from one writing system to their counterparts in another writing system (Daniels & Bright 1996). Some Greek examples are:

- $\psi \Rightarrow ps$
- $\lambda \Rightarrow l$
- $\mu\pi \Rightarrow b$

Frequently, more than one transliteration system exists for the same language pair (the Arabic first name *سعيد* is mainly transliterated in French as *Saïd* but sometimes also as *Said*, *Sayyed* and *Saeed*), which explains why different target language versions may exist for the same name and the same source-target language pair. As pointed out in bullet (h) in Section 5.1, transliteration rules usually differ depending on the target language. It is less common knowledge that transliteration also exists for languages using the same writing system. For example, the name *Bush* will be found in Latvian language as *Bušs*. We deal with intra-script transliteration in the normalisation step described in Section 6. At the JRC, we use transliteration rules for the following scripts:

- Cyrillic (used for Russian, Bulgarian and Ukrainian):
Симеон Маринов → Simeon Marinov;
- Greek: Κώστας Καραμανλής → Kostas Karamanlis;
- Arabic (used for Arabic, Farsi and Urdu; some additional transliteration rules were added for Farsi and Urdu):
جلال طالباني → jlal tlbani (“Jalal Talabani”);
- Devanagari (used for Hindi and Nepalese):
सोनिया गांधी → soniya gandhi.

Transliteration sometimes produces name forms that are rather different from the usual spelling. For frequent names, it is thus appropriate to hard-code the transliteration of the full name. Here are some examples of source language strings, their letter-by-letter transliteration and the aimed-for target language form:

- Russian Джордж → [Djordj] → *George*;
- Russian Джеймс → [Djajms] → *James*;
- Hindi डब्ल्यू → [dableyu] → W (as in *George W. Bush*);

All other spelling differences of transliterated versus non-transliterated names are dealt with in the normalisation step, described in the next section. Adding transliteration tables for new languages using letters (alphabetical scripts) or syllables is not difficult.⁸ Adding rules for Hindi took us two hours. Dealing with ideographic languages such as Chinese is harder and needs different procedures.

6 Rules to identify name variants

In Section 5.1, we have seen a number of reasons why variants exist for the same name. After having applied named entity recognition in up to nineteen languages over a period of about five years, we have found up to 170 variants for the same name.⁹ Identifying these variants as referring to the same person has many advantages, including improved search and retrieval, as well as more accurate results for tasks where person co-references are required such as social network generation based on quotations (Pouliquen et al. 2007a), on co-occurrence (Pouliquen et al. 2007b) or on specific relation information extracted from documents (Tanev 2007). There is thus a clear need for methods to identify whether similar, but different names found in text are variants belonging to the same person or not. There are a number of known approaches to identify name equivalences for specific language *pairs*. In this section, we present an alternative approach, which is language and language pair-independent. It consists of normalising names into an abstract *consonant signature*. This consonant signature can then be used as the basis for comparing all (normalised) names found. All name pairs that have a similarity above a certain threshold will be marked as referring to the same person. The threshold was set so that only good name pairs would be merged for a given test set. The idea behind this is that having two entries for the same person is less harmful than that of merging two different persons. For details on the name normalisation and name merging processes, see Steinberger & Pouliquen (2007).

For every name found in the analysis of news articles in 19 languages carried out daily by the NewsExplorer application, we first check whether the name already has an entry in the NewsExplorer name database. Both main names (aliases) and known name variants are considered. All unknown names will be normalised (see Section 6.1) and compared to the consonant signature of any of the known names. If any of the consonant signatures coincide, name similarity measures will be applied (see Section 6.2). If successful, the new name variant will be added to the existing name. Otherwise, the name will be added as a new name into the database. Names found only ever once are otherwise ignored in order to avoid typos entering the database. If a name is found at least five times, this name gets the status of a frequent name so that it will be found by lookup in any future news articles (see Section 5).

6.1 Language-independent name normalisation rules

The idea behind name normalisation is to create a language-independent representation of the name. In principle, a representation of the (approximate) pronunciation of the name would be a good normalised form, as suggested by the *Soundex* algorithm for English (Hall & Dowling 1980). However, in order to infer the pronunciation of a name, it is necessary to know the original language of the name. In news articles, this information is not normally known as known persons from around the world are being talked about in the newspapers of any other country. To give an example: the name of the previous French president *Chirac* (pronounced in French as /ʃ iʁ ak/) would be pronounced as /kiʁ a k/ if it were an Italian name, as /çir a k/ in German, etc., while the name *Chiamparino* (Mayor of the city of Turin) should be pronounced as /kia mpa r ino/. Automatic identification of the origin of a name (Konstantopoulos 2007) produces moderate results.

In order to overcome this problem, empirical observations on name spelling differences for the same name in many different languages have been used to produce normalisation rules that will be applied to all names, independently of their origin. The following are some examples:

- Name-initial ‘Wl-’ and the name-final ‘-ow’ for Russian names will get replaced by ‘Vl-’ and ‘-ov’. This is to accommodate the typical German transliteration for names like *Vladimir Ustinov* as *Wladimir Ustinow*.

⁸ Various transliteration tables can be found at the *Institute of the Estonian Language* and the *American National Geospatial Agency* (<http://transliteration.eki.ee/>; <http://earth-info.nga.mil/gns/html/romanization.html>).

⁹ The terrorist Abu Musab al-Zarqawi: see <http://press.jrc.it/NewsExplorer/entities/en/285.html>.

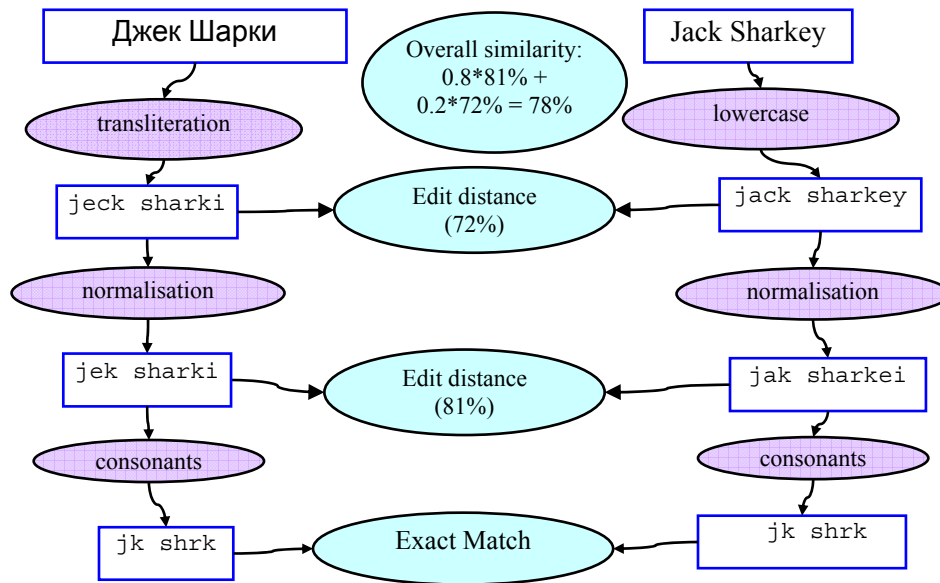


Figure 1. The Figure shows the transliteration and normalisation steps, as well as the similarity measurement applied to a name pair. The two names will not be associated to the same person because the overall similarity is 0.78 and thus lower than the requested threshold of 0.94.

- The Slovene strings 'š', the Turkish 'ş', the German 'sch', the French 'ch' will get replaced by 'sh' in order to neutralize frequent spelling variants such as *Başar al Asad*, *Baschar al Assad*, *Bachar al Assad* and *Başar Esad*.
- The French letter combination 'ou' will get replaced by 'u' to capture the regular transliteration differences for the sound /u/ in names like *Ustinov*, which is spelled in French as *Oustinov*.
- The letter 'x' will get replaced by 'ks', etc.

These normalisation rules are exclusively driven by practical requirements and have no claim to represent any underlying linguistic concept. They are written as a set of regular expression substitutions. For example, the normalisation rule for the Slavic ending *-ski* is written by the substitution rule below and will normalise the following names to a common suffix: *Stravinski*, *Stravinsky* (Finnish), *Stravinskij* (Slovak), *Sztravinszkij* (Hungarian), *Stravinskij* (Icelandic):

$s/sz?k[yií]j?/ski/$ (At the end of a word)

The final normalisation step consists of removing all vowels. Vowels frequently get distorted during transliteration and their removal is compulsory for languages using the Arabic script as short vowels are not normally written in Arabic. An original Russian name such as Джек Шарки will thus go through the stages transliteration (*jeck sharki*), normalisation (*jek sharki*) and vowel removal (*jk shrk*).

6.2 Similarity measure used to compare names with the same normalised form

All names having the same normalised form are considered name variant *candidates*. For each candidate pair, the edit distance similarity measure is applied twice, one time each on two different representations of the name: once between the normalised forms with vowels and once between the lowercased transliterated forms (see **Figure 1**). The two similarities have relative weights of 0.8 and 0.2. By applying the similarity measure only to name pairs with an exact signature match, we miss some good candidates, but the pre-filtering saves a lot of precious computing time. For further details, see Steinberger & Pouliquen (2007).

Both the name normalisation and the similarity calculation steps are applied across the board to all names found in the currently 19 NewsExplorer languages and do not depend on the input language. There is thus no need for language pair-specific training or mapping rules. However, when adding a language with a new script, a new set of transliteration rules needs to be added and – depending on the language – it may be useful to add additional normalisation patterns (which will then be applicable to all languages).

7 Document categorisation using a multilingual thesaurus

In addition to extracting known and new named entities in each news article and cluster, each cluster gets assigned a number of subject domain codes that serve as kind of a subject domain signature. The Eurovoc thesaurus¹⁰ is not particularly well suited for this task, as it was developed to manually index specifically parliamentary and legal documents. Its main advantages are that it exists for many different languages and that there are large collections of manually classified documents.

7.1 Eurovoc Thesaurus

Eurovoc is a hierarchically organised controlled vocabulary that was developed by the European Parliament, the European Commission's Publications Office, together with many national parliaments in and outside the European Union (EU), for the cataloguing, search and retrieval of their large multilingual document collections. Currently, the cataloguing is a manual (intellectual) process. Eurovoc is a steadily growing resource that is currently available in all official EU language versions plus a few more non-EU languages (e.g. Russian).

Its over 6,000 descriptor terms are hierarchically organised into 21 fields and, at the second level, into 127 micro-thesauri. The maximum depth is eight levels. In addition to the approximately 6,000 pairs of broader terms (BT) and narrower terms (NT), there are about 3,000 pairs of related terms (RT) linking descriptors not related hierarchically.

Due to its wide coverage represented by the relatively small number of descriptors, the descriptor terms are mostly rather abstract multi-word expressions that are unlikely to be found verbatim in the texts. Eurovoc examples are PROTECTION OF MINORITIES, FISHERY MANAGEMENT and CONSTRUCTION AND TOWN PLANNING¹¹.

7.2 Mapping Documents to Eurovoc

The procedure of mapping texts written in different languages automatically onto the multilingual Eurovoc thesaurus is described in detail in Pouliquen et al. (2003) so that this process will only be sketched here.

It is not possible to base an automatic Eurovoc thesaurus descriptor assignment on the actual occurrence of the descriptor text in the document because the lexical evidence is weak and even misleading: The descriptor text occurs explicitly in only 31% of documents manually indexed with this descriptor. On the other hand, in approximately nine out of ten documents, the descriptor text is present explicitly even though the descriptor itself had *not* been assigned manually. This means that basing the assignment on the presence of the descriptor text in the document would lead to wrong results in the majority of cases.

For this reason, we have taken another, fuzzy approach. For each descriptor, we automatically produce its profile, i.e. a ranked set of words that are statistically related to the descriptor so that we have more lexical evidence at hand that indicates that a certain descriptor should be assigned to a text. As these words are statistically, but not always semantically related, we refer to these pertinent words as descriptor *associates*. **Table 1** shows the top of the relevance-ranked associate list for the descriptor TRANSPORT OF DANGEROUS GOODS.

We produce these associate lists on the basis of a large collection of manually indexed documents (the *training set*), by comparing the word frequencies in the subset of texts that have been indexed manually with a certain descriptor with the word frequencies in the whole training set. For the comparison, we use a combination of Dunning's statistical log-likelihood test (or G^2) to reduce the number of words to be considered (dimensionality reduction) with filters and various IDF (Inverse Document Frequency) weightings (Salton & Buckley 1988). This method automatically produces lists of typical words for each descriptor. It also produces information on the degree with which these words are typical.

Lemma	Weight
dangerous_goods	33
radioactive_material	19
by_road	19
carriage	19
dangerous	18
plutonium	17
radioactive_waste	15
nuclear_fuel	15
shipment	15
adr	14
bind_for	13
tank	13
receptacle	13
transport	13
pollute	12
nuclear_waste	12

Table 1. Most important English associated lemmas for Eurovoc descriptor TRANSPORT OF DANGEROUS GOODS.

¹⁰ See <http://europa.eu/eurovoc/>.

¹¹ We write all Eurovoc descriptors in small caps.

We make use of a generous list of stop words to avoid that less meaningful words have an impact on the categorisation performance. Experiments have shown (Pouliquen et al. 2003) that a good stop word list has a big impact on the categorisation performance. We carried out a number of further experiments to establish whether normalising the texts before applying the word weighting and the categorisation steps would be useful, i.e. lemmatising all words and marking up frequent multi-word terms. However, the performance gain was minimal. As lemmatisation software is hard to get by for 19 or more languages, we only use stop word lists and do not apply any

Rank	Descriptor	Cosine
1	VETERINARY LEGISLATION	42.4%
2	PUBLIC HEALTH	37.1%
3	VETERINARY INSPECTION	36.6%
4	FOOD CONTROL	35.6%
5	FOOD INSPECTION	34.8%
6	AUSTRIA	29.5%
7	VETERINARY PRODUCT	28.9%
8	COMMUNITY CONTROL	28.4%

Table 2. Assignment results (top 8 descriptors) for a document found on the internet ('Food and veterinary Office mission to Austria')¹²

linguistic pre-processing as part of Eurovoc-indexing in NewsExplorer.

When we want to index a new text automatically with Eurovoc descriptors, we make a statistical comparison of the frequency list of its words with the associate lists of all descriptors to check which associate lists are most similar to the text's word list. The most similar associate lists, according to some statistical similarity measure, indicate the most appropriate descriptor terms. The result is thus a long ranked list of Eurovoc descriptors assigned to this document. Typically, we keep the highest-ranking 100 descriptors. The example in **Table 2** shows the assignment results of a document found on the internet¹².

To speak with text classification terminology (Sebastiani, 1999), the mapping of texts onto the Eurovoc thesaurus is a category ranking classification task using a machine learning approach, where an inductive process builds a profile-based classifier by observing the manual classification on a training set of documents with only positive examples. For each Eurovoc descriptor, we built a category profile consisting of a set of words and their weight. Unlike usual classifiers, our system does not decide against the appropriateness of a class. Instead, it produces long ranked lists of more or less relevant classes for each document. This representation is more suitable for document similarity calculation. According to Sebastiani (1999), the k-nearest-neighbour (KNN) approach tends to produce best results. Although we have not tried this approach, it does not seem computationally viable to apply it to our training set of tens of thousands of documents, and we doubt that it would be the most appropriate technique for our multi-class categorisation problem.

The method used is language-independent. It has been trained for 22 official EU languages, using the JRC-Acquis multilingual parallel corpus (Steinberger et al. 2006).

The system was evaluated for English and Spanish on a training set consisting of 30231 texts, and using a complementary test set of 590 representative, but randomly chosen texts (for details, see Pouliquen et al. 2003). The descriptor assignment was evaluated against the annotation of two separate indexing professionals. The precision achieved for the eight highest-ranking descriptors assigned automatically (eight was the average number of descriptors assigned manually) was 67% (F=65). This is 86% as good as the inter-annotator agreement of 78% (67/78=86%). These results are not perfect, but they were good enough for the Spanish Congress of Deputies in Madrid to decide to use the software interactively in their daily Eurovoc-indexing process. The software has been used in Madrid since 2006.

8 Topic detection and tracking

This section describes the NewsExplorer functionality to link related news clusters over time (monolingual topic tracking) and across languages (cross-lingual topic tracking). For the cross-lingual part of the task, it is necessary to calculate the similarity of documents across languages. For further details on this work, see Pouliquen et al. 2008.

8.1 Related work

Topic Detection and Tracking (TDT) was promoted and meticulously defined by the US-American DARPA programme (see Wayne 2000). An example explaining the TDT concept was that of the Oklahoma City bombing in 1995, where not only the bombing, but also the related memorial services, investi-

¹²http://europa.eu.int/comm/food/fs/inspections/vi/reports/austria/vi_rep_oste_1074-1999_en.html

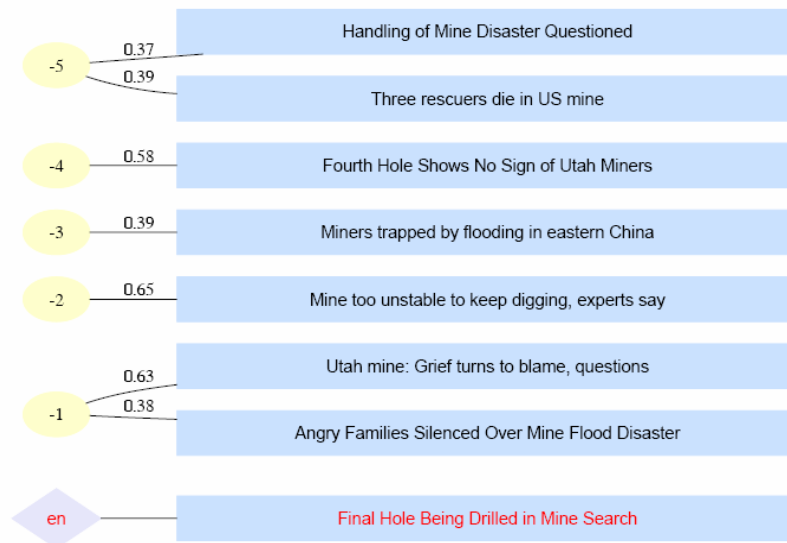


Figure 2. Example of historical links between clusters: The graph shows the cosine similarity between today’s English language cluster (*Final hole being drilled*) and seven clusters identified during five previous days. Only clusters with a similarity above 0.5 will be retained.

gations, prosecution etc. were supposed to be captured. Human evaluators will often differ in their opinion whether a given document belongs to a topic or not, especially as ‘topic’ can be defined broadly (e.g. the Iraq war and the following period of insurgency) or more specifically. For instance, the capture and prosecution of Saddam Hussein, individual roadside bombings and air strikes, or the killing of Al Qaeda leader Abu Musab al-Zarqawi could either be seen as individual topics or as part of the Iraq war. This fuzziness regarding what is a ‘topic’ makes a formal evaluation rather difficult.

Since 2000, the TDT task was part of the TIDES programme (Translingual Information Detection, Extraction and Summarisation), which focused on cross-lingual information access. The goal of TIDES was to enable English-speaking users to access, correlate and interpret multilingual sources of real-time information and to share the essence of this information with collaborators. The purpose of our work includes the topic detection and tracking as well as the cross-lingual aspect. A difference between our own work and TIDES is that we need to monitor more languages and that we are interested in all cross-lingual links, as opposed to targeting only English.

All TDT and TIDES participants used either Machine Translation (MT; e.g. Leek et al. 1999) or bilingual dictionaries (e.g. Wactlar 1999) for the cross-lingual tasks. Performance was always lower for cross-lingual topic tracking (Wayne 2000). An interesting insight was formulated in the “native language hypothesis” by Larkey et al (2004), which states that topic tracking works better in the original language than in (machine)-translated collections. Various participants stated that the usage of named entities helped (Wayne 2000). Taking these insights into account, we always work in the source language and make intensive use of named entities.

Outside TDT, an additional two approaches for linking related documents across languages have been proposed, both of which use bilingual vector space models: Landauer & Littman (1991) used bilingual *Lexical Semantic Analysis* and Vinokourov et al. (2002) used *Kernel Canonical Correlation Analysis*. These and the previously mentioned approaches have in common that they require bilingual resources and are thus not easily scalable for many language pairs. For N languages, there are $N * (N - 1) / 2$ language pairs (e.g. for 20 languages, there are 190 language pairs and 380 language pair directions). Due to the multilinguality requirement in the European Union (EU) context (23 official EU languages as of 2007), Steinberger et al. (2004) proposed to produce an interlingual document (or document cluster) representation based on named entities (persons, organisations, disambiguated locations), units of measurement, multilingual specialist taxonomies (e.g. medicine), thesauri and other similar resources that may help produce a language-independent representation of documents.

Similarly to Steinberger et al. (2004), the work presented here also goes beyond the language pair-specific approach, but it does not make use of all the proposed information types. It also goes beyond similar work by Pouliquen et al. (2004) who link individual news clusters over time and across languages, but without aggregating the news clusters into stories. The novel part of our work is that we aggregate

Lang.	Biggest title	Keywords
En	US Airways won't pursue Delta forever	<i>United states</i> / Doug Parker, Delta Airlines / airways, offer, emerge, grinstein, bid, regulatory, creditors, bankruptcy, atlanta, increased
It	Stop al massacro di balene. Il mondo contro il Giappone	<i>Australia, New Zealand, Japan</i> / Greenpeace International, John Howard / caccia, megattere, balene, sydney, acqua, mesi, antartico, salti
Es	Mayor operación contra la pornografía infantil en Internet en la historia de España	Guardia Civil, Fernando Herrero Tejedor / pornografía, imputados, mayor, cinco, delito, internet, registros, siete, informática, sci
De	Australian Open: "Tommyator" mit Gala-Vorstellung	<i>Russia, Australia, United states</i> / Australian Open, Mischa Zverev / satz, tennis, deutschen, bozoljač, erstrunden, melbourne, kohlschreiber, Donnerstag
Fr	Il faut aider l'Afrique à se mondialiser, dit Jacques Chirac	Jacques Chirac, African Union / afrique, sommet, continent, président, cannes, darfour, état, pays, conférence, chefs, omar

Table 3: Examples of stories, their biggest titles and their corresponding keywords. Countries are displayed in italic, person and organisation names in boldface.

news clusters into more compact and high-level representations (the *stories*), by exploiting the monolingual and cross-lingual cluster links and by adding additional filtering heuristics to eliminate wrong story candidate clusters. As a result, long-term developments can be visualised and users can explore the development of events over long time periods by exploiting the automatically established links. This aggregation allows to automatically compile meta-information for each story, including article and cluster statistics as well as lists of entities mentioned in the story.

Compared to commercial or other publicly accessible news analysis and navigation applications, the one presented here is unique in that it is the only one offering automatic linking of news items related either historically or across languages. The news aggregators *Google News* (<http://news.google.com/>) and *Yahoo! News* (<http://news.yahoo.com/>), for instance, deliver daily news in multiple languages, but do not link the found articles over time or across languages. The monolingual English language applications *DayLife* (<http://www.daylife.com/>), *SiloBreaker* (<http://www.silobreaker.com/>), and *NewsVine* (<http://www.newsvine.com/>) do not link related news over time either. *NewsTin* (<http://www.newstin.com/>) is the only one to offer more languages (eleven) and to categorise news into a number of broad categories, but they, again, do not link related news over time or across languages.

Regarding cross-lingual document linking outside the domain of daily news aggregation and exploration, our application differs from state-of-the-art work in that it does not rely on bilingual resources, but uses an interlingual document representation allowing to add on new languages easily.

8.2 Linking clusters over time – building stories

For each language separately and for each individual cluster of the day, we compute the similarity with all clusters of the past 7 days (see **Figure 2**). Similarity is based on the keywords associated with each cluster. If the *cosine* similarity between the keyword vectors of two clusters is above the empirically derived threshold of 0.5, clusters are linked. A cluster can be linked to several previous clusters, and it can even be linked to two different clusters of the same day.

8.2.1 Building the stories

Stories are composed of clusters. If a new cluster is similar to clusters that are part of a story, it is likely that this new cluster is a continuation of the existing story. For the purpose of building stories, individual and yet unlinked clusters of the previous seven days are assumed to be (single cluster) stories. If clusters have not been linked to within seven days, they remain individual clusters that are not part of a story. Building stories out of clusters is done using the following incremental algorithm:

```
// For a given day
for each cluster c
  for each story s score[s]=0;
  for each cluster cp (linked to c)
    if (s: story containing cp) then
      score[s] += (1-score[s])*similarity(cp, s);
    endif
  endfor
  if (s: story having the maximum score) then
    add c to story s
  else // not similar to any existing story
    create new story containing only c
```

This Week's New Stories	This Month's New Stories	Biggest Stories
Heathrow hassle continues for third day March 28, 2008 - March 31, 2008	Fed cuts short-term interest rate March 14, 2008 - March 27, 2008	At least 40 die in Israeli attack on Qana December 5, 2005 - November 3, 2006
GMS transport corridors be turned to economic ones: PM Dung March 28, 2008 - March 31, 2008	Spitzer steps down as New York governor March 10, 2008 - March 19, 2008	Abbas suspends peace talks with Israel April 21, 2007 - March 31, 2008
Report: North Korea test-fires missiles March 28, 2008 - March 31, 2008	Bush heads to his last NATO summit with eye on Russia March 18, 2008 - March 31, 2008	Iran welcomes US downgrading of nuclear threat December 2, 2006 - March 30, 2008
Bush and Australian prime minister urge China to meet with Dalai Lama March 27, 2008 - March 31, 2008	Astronauts head out to build 12-foot, 3,400-pound robot March 14, 2008 - March 27, 2008	Saddam Hussein Death sentence Death penalty for Saddam Hussein December 1, 2005 - December 4, 2006

Figure 3: Examples of English language stories as they appeared on the NewsExplorer main page as of 2 April 2008.

```
endif
endfor
```

When deciding whether a new cluster should be part of an existing story, the challenge is to combine the similarities of the new cluster with each of the clusters in the story. As stories change over time, the new cluster is only compared to the story's clusters of the last 7 days. In the algorithm to determine whether the new cluster is linked to the story, the similarity score is computed incrementally: The score is the similarity of the new cluster with the latest cluster of the story (typically yesterday's) plus the similarity of the new cluster with the story's cluster of the day before multiplied with a reducing factor ($1 - score_{i-1}$), plus the similarity of the new cluster with the story's cluster of yet another day before multiplied with a reducing factor ($1 - score_{i-2}$), etc. The reducing factor helps to keep the similarity score between the theoretical values 0 (unrelated) and 1 (highly related):

$$score_i = \begin{cases} 0 & (i = 0) \\ (1 - score_{i-1}) \cdot sim(c_i, s) & (0 < i < 7) \end{cases}$$

If the final score is above the threshold of 0.5, the cluster gets linked to the existing story. Otherwise it remains unlinked. This algorithm is run every day (in sequential order) in the following nine languages (core languages): Dutch, English, French, German, Italian, Portuguese, Slovene, Spanish and Swedish. The story building algorithm is language-independent, but as it is computationally heavy, it is currently only applied to nine of the 19 NewsExplorer languages.

Out of the daily average of 970 new clusters (average computed for all nine languages over a period of one month), only 281 get linked to an existing story (29%) and 90 contribute to a new story (9%). The remaining 599 clusters (62%) remain unlinked singleton clusters. A small number of stories are very big and go on over a long time. This reflects the big media issues such as the Iraq insurgence, the Israel-Palestine conflict and the Iran-nuclear negotiations. The latter is the currently longest story ever (see <http://press.jrc.it/NewsExplorer/storyedition/en/TurkishDailyNews-9bb70c1ffbe0b89cccf50f77b28de0db.html>).

8.2.2 Aggregating and displaying information about each story

For each story, daily updated information gets stored in the NewsExplorer knowledge base. This includes (a) the title of the first cluster of the story (i.e. the title of the medoid article of that first cluster); (b) the title of the biggest cluster of the story (i.e. the cluster with most articles); (c) the most frequently mentioned person names in the story (*related people*); (d) the person names most highly associated to the story (*associated people*, see below); (e) the most frequently mentioned other names in the story (mostly organisations, but also events such as *Olympics*, *World War II*, etc.); (f) the countries most frequently referred to in the story (either directly with the country name or indirectly, e.g. by referring to a city in that country); (g) a list of keywords describing the story (see below). This meta-information is exported every day into XML files for display on NewsExplorer. The public web pages display up to 13 keywords, including country names and two person or organisation names (see **Table 3**). For a full list of meta-information about stories, see the NewsExplorer pages. Stories are currently accessible through three different indexes (see **Figure 3**): the stories of the week, the stories of the month and the biggest stories (all displayed on the main page of NewsExplorer). The biggest stories are ordered by the number of clusters they contain without any consideration of the beginning date or the end date. The stories of the month present stories that started within the last 30 days, stories of the week those that started within the last seven days.

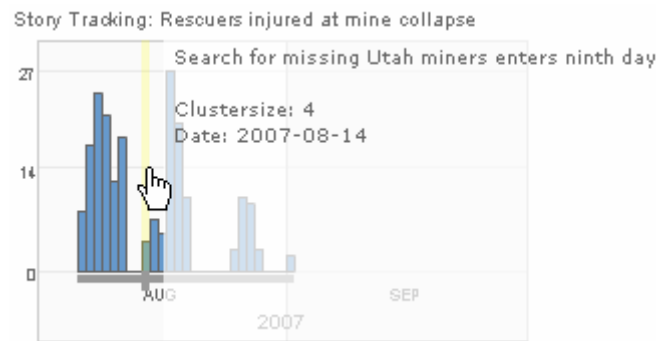


Figure 4. Sample of a short story timeline. When mousing over the graph, title, date and cluster size for that day are displayed. A simple click allows to jump to the relevant cluster, enabling users to explore the story.

For each story, a time line graph (a flash application taking an XML export as input) is produced automatically, allowing users to see trends and to navigate and explore the story (**Figure 4**)¹³. While a story can have more than one cluster on a given day, the graph only displays the largest cluster for that day.

The story's keyword signature is computed using the sum of keywords of each constituent cluster. If any of the keywords represents a country, it will be displayed first. A filtering function eliminates keywords that are part of one of the selected entities displayed. For instance, if a selected entity is *George W. Bush* and a selected country is *Iraq*, the keywords *Bush*, *George*, *Iraqi*, etc. will not be displayed.

8.3 Cross-lingual cluster and story linking

For each daily cluster in NewsExplorer's nine core languages, the similarity to clusters in the other 18 languages is computed. To achieve this, we produce three different language-independent vector representations for each cluster (for further details, see Pouliquen et al. 2004): a weighted list of Eurovoc subject domain descriptors (*eurov*, available only for EU languages, see Section 7), a frequency list of person and organisation names (*ent*, see Section 4), and a weighted list of direct or indirect references to countries (*geo*, see Section 3). As a fourth ingredient, we also make use of language-dependent keyword lists because even monolingual keywords sometimes match across languages due to cognate words (*cog*), etc. (e.g. *tsunami*, *airlines*, *Tibet* etc.). The overall similarity *clsim* for two clusters c' and c'' in different languages is calculated using a linear combination of the four cosine similarities, using the intuitively set values for α, β, γ & λ to 0.4, 0.3, 0.2 and 0.1, respectively (see **Figure 5**):

$$clsim(c', c'') = \alpha \cdot eurov(c', c'') + \beta \cdot geo(c', c'') + \gamma \cdot ent(c', c'') + \lambda \cdot cog(c', c'')$$

8.3.1 Filtering and refining cross-lingual cluster links

The process described in the previous paragraphs produces some unwanted cross-lingual links. We also observed that not all cross-lingual links are transitive although they should be. We thus developed an additional filtering and link weighting algorithm to improve matters, whose basic idea is the following: When clusters are linked in more than two languages, our assumption is: If cluster A is linked to cluster B and cluster C, then cluster B should also be linked to cluster C. We furthermore assume that if cluster B is not linked to cluster C, then cluster B is less likely to be linked to cluster A. The new algorithm thus checks these 'inter-links' and calculates a new similarity value which combines the standard similarity (described in Section 8.3) with the number of inter-links. The formula punishes links to an isolated cluster (i.e. links to a target language cluster which itself is not linked to other linked languages) and raises the score for inter-linked clusters (i.e. links to a target language cluster which itself is linked to other linked languages). The new similarity score uses the formula:

$$sim'(S', S'') = sim(S', S'') \cdot \frac{CI(S')}{\sqrt{EI(S')}}$$

¹³ Available on page <http://press.jrc.it/NewsExplorer/storyedition/en/guardian-ee9f870100be631c0147646d29222de9.html>

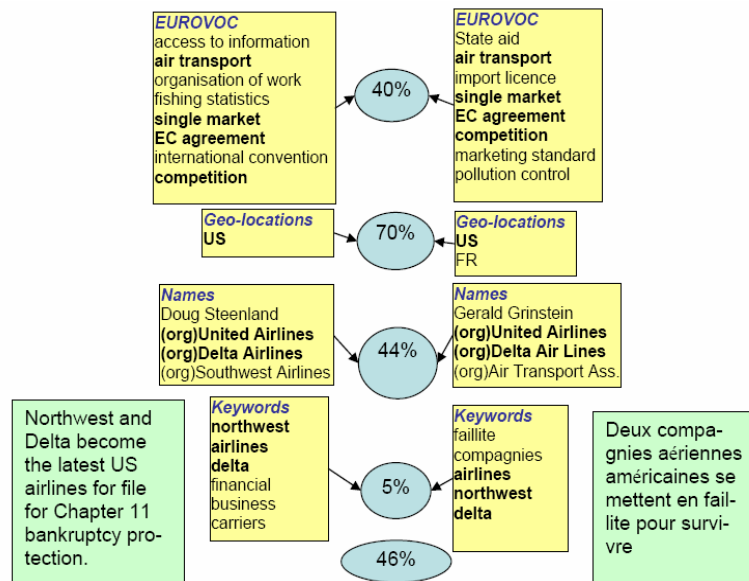


Figure 5. Similarity calculation for an English and a French cluster. The overall similarity for these two clusters, based on the linear combination of four different vectors, is 0.46.

with $CI(S)$ being the number of computed cross-lingual links and $EI(S)$ being the number of expected cross-links (i.e. all cross-language links observed when looking at all languages). For instance, if a cluster is linked to three languages and these are linked to a further three, then $CI(S')=3$ and $EI(S')=6$.

In Pouliquen et al. (2008), we present work on linking whole stories across languages, but the results of this work are not yet available online.

9 Quotation recognition

For each quotation (reported speech) found in the news articles, NewsExplorer attempts to identify automatically the person who made the quotation and, if applicable, any entity referred to inside the quotation. This work has been described in detail in Pouliquen et al. (2007a).

9.1 Method

It was our aim to detect quotations in many different languages. We therefore kept the linguistic input as simple as possible, by mainly relying on lexical patterns with character-level regular expressions, as these are easily transposable to new languages. While we are aiming at detecting quotations in all NewsExplorer languages, we currently detect them in only eleven of them (Arabic, German, English, Spanish, French, Italian, Dutch, Portuguese, Romanian, Russian and Swedish).

The method used is quite simple: we look in the text of each article for quotation markers that are found close to reporting verbs (*say, declare* etc.) and known person names. Known person names are those that have been found in at least five different NewsExplorer news clusters (see Section 4).

In most news articles, names found next to quotes are not full names consisting of first and last name. Common example types for quotations found in text are the following:

1. Tony Blair said "We stand ready to support you in every way".
2. "We stand ready to support you in every way," Blair said.
3. Tony Blair visited Iraq... He said "We stand ready to support you in every way".
4. Tony Blair visited Iraq... "We stand ready to support you in every way" the British Prime Minister said.

Our system currently only captures the first two types. Example (1) is not very common because the newspapers usually first talk about the context (*Tony Blair visiting Iraq*) and only then they introduce quotes. Example (2) is more common and it is relatively easy to detect accurately. The issue here is that only the last name is mentioned and that we have to infer that the quote is by *British Prime Minister Tony Blair* even though there may be other persons with the name of 'Blair' in our database. We achieve this by first

scanning the text for all occurrences of full names (consisting of first *and* last name), and by then assuming a co-reference between the full name and the name part found. In order to recognise the person doing the quoting in the third example, we would need to identify that the pronoun *he* refers to *Tony Blair*. We do not currently attempt to resolve such cases of anaphora usage. We also do not currently try to identify the co-reference between ‘British Prime Minister’ and ‘Tony Blair’ in cases like (4), but have plans to do so.

9.2 Algorithm for quote recognition

We aim to detect only those quotations that are accompanied by a named person because we cannot think of a use for anonymous quotations or for those where we do not know the name of the speaker (e.g. “said their neighbour”). The system will recognise quotations only if it successfully detects three parts: the speaker name, a reporting verb and the quotation.

Our analysis of quotations in the news in various languages showed that many of the quotations are similar to the two examples below, i.e. the person making the quotation is either mentioned immediately before or after the quotation:

“I don’t think Congress ought to be running the war,” Bush said yesterday.

Mr. Wolfowitz said yesterday “I will accept any remedies”.

What complicates matters is the use of anaphoric expressions instead of person names (‘he said’, ‘added the President’) and the fact that modifiers such as yesterday or in a radio interview may be found between the reporting verb and the quote. While we do not currently deal with anaphoric expressions at all, we do try to capture at least some modifiers.

9.3 Components for quotation recognition

Most quotations can be identified using a small number of rules. Our rules, described in Section 9.4, make use of the components described in paragraphs (A) to (F):

- (A) quotation marker identification (quote-characters like “, ”, «, » etc.)
- (B) reporting verbs (e.g. *confirmed, says, declared* ...)
- (C) general modifiers, which can appear close to the verb (e.g. the adverb *yesterday*)
- (D) determiners, which can appear between the verb and the person name (e.g. *the*)
- (E) trigger-for-person (e.g. *British Prime Minister*)
- (F) person name (e.g. *Tony Blair*)
- (G) a list of matching rules (e.g. *name verb [adverb] quote-mark QUOTE quote-mark*)

We will now discuss these in detail.

(A) Quotation markers

In order to mark the quotation itself, we first identify and normalise the following quote-marks: ["] (two single apostrophes), [`] (two curly apostrophes), [,] (bottom quotes, used in German and some Dutch newspapers), [« /.../ »] (French quotes), [“ /.../ ”] (the English curly quotes), [<< /.../ >>] (two angled brackets), ["/.../"] (double single-quotes), [‘ /.../’] (single quotes)

(B) Reporting verbs

They define a verb or any of its inflections that express that the string between quote-marks is a quotation. Without the presence of any of these verbs, we will not recognise the quotation. Examples are English *says, said, added, commented, sums up* and Italian *ha detto, dice, diceva*.

(C) General modifiers

These consist of quite generous lists of strings or regular expressions that are allowed before or after the verb. These strings are generally adverbs (often, also, today...), but there are also some compound expressions (*on television, last month*)¹⁴. We do not make use of external dictionaries, part-of-speech taggers or syntactic patterns. Instead, the list of modifiers has been derived empirically. To avoid listing all forms of verbs (*have said, might have said, would say*...), we also included auxiliary and modal verbs in this list of modifiers (in English: *has, have, had, would, might, could, do, did, does*).

¹⁴ The Spanish configuration includes the following regular expression (*por la |en la |a la |en*)(*mañana|tarde*) recognising *por la mañana* or *a la tarde*. In French: *pour sa part* and even the days of the week (*lundi, mardi*...) as it is quite common to say in French: “...” *a dit lundi Jacques Chirac*.

(D) Determiners

In some cases, determiners can precede the name of a person. In our rules, they are allowed between the verb and the person name (English: the, French: le, un, l', German: der, die, seine).

(E) Trigger-for-person

As described in Section 4.1, these patterns are usually titles of persons (*Dr., Prime Minister, French President...*), expressions referring to nationality (e.g. *the Iranian*), and more. In a random set of 240 English quotations, we found that in nine cases (3.75%) the title of the person was found before the person name. This low number is presumably due to the fact that the titles are used when the person is first introduced while quotes are usually mentioned further down in the article. For the detection of names in NewsExplorer, we built (semi-automatically) an extensive list of such trigger words. In English, the list currently comprises more than 1,000 items. Recognition patterns also allow for combinations of several of them (e.g. *young Spanish Ambassador*).

(F) Person name

The quotation recognition tool makes use of the names identified by the named entity recognition tool described in Section 4. The person names are marked up in each article. In order to resolve the name part co-reference resolution, we then look up in text the uppercase words that are also part of a full name found elsewhere in the text. This method can identify 'Tony Blair' as the author even if only the last name of the author is used in the text (e.g. *[Tony Blair] visited Iraq yesterday. ... "I reiterate our determination to stand four-square behind you" said [Blair]*).

9.4 Matching rules

In our survey of the various ways to express a quotation across languages, we found three generic rules and a number of additional language-specific rules. The three generic rules are:

- (1) *quote-mark QUOTE quote-mark [,] verb [modifier] [determiner] [title] name*
e.g. *"blah blah", said again the journalist John Smith.*
- (2) *name [, up to 60 characters ,] verb [[:that] quote-mark QUOTE quote-mark*
e.g. *John Smith, supporting AFG, said: "blah blah".*
- (3) *quote-mark QUOTE quote-mark [; or ,] [title] name [modifier] verb*
e.g. *"blah blah", Mr John Smith said.*

The following format was found only in Italian and Russian articles:

- (4) *quote-mark QUOTE1 - [modifier] verb name - QUOTE2 quote-mark*
e.g. *"Ciampi – ha detto Berlusconi – ha favorito la sinistra perché era un uomo della sinistra"*
where the author (here Berlusconi) and the reporting verb (*said*) is included *inside* the quotation marks, marked by hyphens.

The Swedish convention for quotations includes sentences beginning with one or two hyphens "--":

- (5) *-- QUOTE, verb [adverb] [title] name*
e.g. *-- Vi försökte uppmuntra samverkan, säger Urban Lundmark.*

A specific Arabic pattern is to mention the verb *before* the person name. We therefore introduced the rule:

- (6) *verb [title] name [modifier] quote-mark QUOTE quote-mark*
[and said minister of justice Saddam Hussein to Israel radio "we don't .."]
وقال وزير العدل صدام حسين لإذاعة إسرائيل
"إننا نحمل عباس المسؤولية النهائية عما يحدث"

A switch in the program, activated by information stored in the language-specific parameter file, ensures that such language-specific rules only get applied in the appropriate languages.

9.5 Evaluation and discussion

Users can consult the quotations of each person in NewsExplorer. The process gathers an average of 2,665 quotes per day (1647 of which are found in 7000 English articles every day). As of June 2008, we had a repository of about 2,500,000 quotes, gathered during 3 years of analysis. This repository is not currently fully exploited apart from displaying quotations of/about a person as part of the NewsExplorer's person pages. From an application-oriented point of view, this works rather well: For many persons, NewsExplorer displays recent quotes from or about the person in many different languages.

According to the evaluation described in Pouliquen et al. (2007a), the recall for the recognition of those English language quotes we attempt to identify (i.e. the speaker's name must be mentioned) was 54% at article level. This is relatively low, but due to the redundancy of the news data in NewsExplorer, recall at cluster level in the test set was 100%. This means that identical quotations to those missed in some articles were found in other articles of the same cluster. This result confirms that – within the context of NewsExplorer – it is useful to focus on precision rather than recall. Recall is usually rather good due to the large amount of news data processed. The main reason why some quotations were not recognised is that the modifiers separating the speakers or reporting verbs from the quotation markers were not in our modifier list (e.g. *in a short statement* and *with relief*). Precision was 87.5%. The only error of the system was that it only recognised the first part of a quote, while the second quote after an interruption was not recognised (e.g. *“I'm really happy for Fabio,” Materazzi told the Apcom news agency Friday. “I feel part of this ...”*). A second, mixed-language evaluation (ten languages), showed that 81.7% of the quotations were correctly identified. In 17.5% of cases, the continuation of a quote was missing. In 0.8% (one case), the quotation was assigned to the wrong speaker.

Taking into account the simplicity of the approach, we consider the overall results to be rather good. Precision is rather high, and the relatively low recall at document level is often compensated by the data redundancy, i.e. the same quotation will frequently be found in another news article.

Obvious restrictions of the approach are that (a) there is no co-reference resolution for pronouns and for titles, (b) unknown modifiers that separate the reporting verb and the quotation are not identified and (c) quotes in genitive constructions are currently assigned to the wrong person (In *“...” said Blair's spokesperson*, Blair would be identified as the author of the quote).

However, the simplicity of the system also has important advantages: (a) The process is fully automatic, fast and can detect a high number of quotations in only a few seconds and (b) due to its simplicity, it was possible to adapt the tool quickly to many languages (currently eleven). As time and source of the quotation are always identified and shown in NewsExplorer, users can always read the full article to verify the correctness of the quotation.

10 Conclusion

This document summarises the main functionality and most of the components of the freely accessible news gathering, aggregation and exploration system NewsExplorer. Due to time and space restrictions, related work and state-of-the-art studies have not always been included, but the dedicated publications referred to in each of the sections of this document do contain such information.

Since the very beginning of its existence, it was clear that NewsExplorer had to cover many languages, and ideally all official EU languages. In the course of the years, it turned out that users were also interested in at least some non-EU languages. This multilinguality requirement shaped the main design guidelines used in NewsExplorer, which are above all: “keep it simple”, “try to avoid language-specific resources as much as possible” and – where their usage could not be avoided – “keep language-specific resources separately from the language-independent rules, in separate language-specific parameter files”. In the course of the years, while adding on more and more languages and learning about language-specific issues, we learnt how tricky it can be to adhere to these guidelines. The forthcoming book chapter *Using language-independent rules to achieve high multilinguality in Text Mining* (Steinberger et al, forthcoming) summarises our experiences.

NewsExplorer still does not cover all 23 official EU languages (and will probably never cover all of them, as priorities lie elsewhere), but the currently 19 languages show at least that such high multilinguality is – in principle – possible.

The simplicity of the NewsExplorer tools has its cost: Most of the achieved results are not bad at all and are mostly considered by the users as acceptable, but evaluation exercises show that the tools still make mistakes and could be improved. Presumably, performance could be better if we made use of part-of-speech information (and syntactic parses?) in information extraction rules, applied additional machine learning steps, etc. The reason why we have not invested energy in exploring these possibilities is that development time and linguistic resources are limited and the idea of then having to apply them to all 19 languages is off-putting.

NewsExplorer processes a large number of news articles from many different sources. The data is thus highly redundant. For this reason, the individual text analysis applications can focus on precision rather than recall. The data redundancy frequently compensates for information that was not successfully extracted from individual articles.

Most NewsExplorer tools have been implemented in PERL. For a better integration with the other EMM applications *NewsBrief* and *MedISys* (see <http://press.jrc.it/overview.html>), the tools are slowly being converted to the Java programming language. This is an opportunity to rethink the design of the tools and to tidy up the code. In a couple of years, the inner working of NewsExplorer and its components may therefore be rather different.

Acknowledgements

NewsExplorer development started over four years ago. In the course of the years, many people contributed and helped with their language-specific knowledge. This summary report is an opportunity to thank them all for their effort. The following list of contributors is not exhaustive.

NewsExplorer would not be what it is if the *Europe Media Monitor* core engine did not provide the tens of thousands of tidy and uniform news articles every day, and if web developers did not help to put the application online via robust web interfaces. Thanks for this rich resource and the web development work go to the team leaders Erik van der Goot and Clive Best, as well as to the JRC's whole *Web Mining and Intelligence* Team, which includes Martin Atkinson, Ken Blackler, Jonathan Crawley, Flavio Fuart, Teófilo Garcia, Sunny George Jacob, David Horby, Tamara Öllinger, Monica de Paola and Alastair Wilcox. We also thank our Unit Head Delilah Al Khudhairi for her support.

A number of computational linguists contributed with tools, know-how and effort, including Olivier Deguernel, Tomaž Erjavec, Johan Hagman, Marco Kimler, Mladen Kolar, Jakub Piskorski, Hristo Tanev, Jan Žižka and – above all – Camelia Ignat, who certainly had a lot of influence on how NewsExplorer developed.

These and a number of further colleagues and visitors helped us to adopt NewsExplorer to new languages (e.g. by providing the language-specific parameter files), i.e. Jenya Belyaeva, Ann-Charlotte Forslund, Andrea Heyl, Emilia Käsper, Pinar Özden-Wennerberg, Helen Salak, Irina Temnikova, Anna Widiger, Bart Wittebrood and Wajdi Zaghouni. Finally, we would like to thank our highly multilingual colleague Jenya Belyaeva for her additional help with testing and evaluating the systems and for monitoring the output quality.

References

- Best, Clive, Erik van der Goot, Ken Blackler, Teófilo Garcia & David Horby (2005). *Europe Media Monitor – System Description*. EUR Report 22173-En, Ispra, Italy.
- Bronstein I. N., K. A. Semendjajew, G. Musiol & H Muhlig (1999). *Taschenbuch der Mathematik* (4. ed.). Frankfurt am Main, Thun: Verlag Harri Deutsch.
- Daniels Peter T. & William Bright (eds.) (1996). *The World's Writing Systems*. Oxford University Press, Oxford, UK.
- Hall P. and G. Dowling (1980). *Approximate string matching*. *Computing Surveys*, 12:4, pp. 381-402.
- Konstantopoulos Stasinou (2007). *What's in a name? quite a lot*. In Proceedings of the RANLP workshop 'Workshop on Computational Phonology'. Borovets, Bulgaria.
- Landauer Thomas & Michael Littman (1991). *A Statistical Method for Language-Independent Representation of the Topical Content of Text Segments*. Proceedings of the 11th International Conference 'Expert Systems and Their Applications', vol. 8: 77-85. Avignon, France.
- Larkey Leah, Fangfang Feng, Margaret Connell, Victor Lavrenko (2004). *Language-specific Models in Multilingual Topic Tracking*. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp 402-409.
- Leek Tim, Hubert Jin, Sreenivasa Sista & Richard Schwartz (1999). *The BBN Crosslingual Topic Detection and Tracking System*. In 1999 TDT Evaluation System Summary Papers.
- Leidner Jochen (2007). *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names*. Ph.D. thesis, School of Informatics, University of Edinburgh, Edinburgh, Scotland, UK.
- Mikheev A., M. Moens & C. Gover (1999). *Named Entity Recognition without Gazetteers*. In Proceedings of EACL, Bergen, Norway.
- Piskorski Jakub, Karol Wieloch, Mariusz Pikula & Marcin Sydow (2008). *Towards Person Name Matching for Inflective Languages*. In: Proceedings of the WWW'2008 workshop 'Natural Language Processing Challenges in the Information Explosion Era'. Beijing, China.
- Pouliquen Bruno & Ralf Steinberger (in print). *Automatic Construction of Multilingual Name Dictionaries*. In: Cyril Goutte, Nicola Cancedda, Marc Dymetman & George Foster (eds.): *Learning Machine Translation*. MIT Press.

- Pouliquen Bruno, Marco Kimler, Ralf Steinberger, Camelia Ignat, Tamara Oellinger, Ken Blackler, Flavio Fuart, Wajdi Zaghouni, Anna Widiger, Ann-Charlotte Forslund, Clive Best (2006). *Geocoding multilingual texts: Recognition, Disambiguation and Visualisation*. Proceedings of the 5th International Conference on Language Resources and Evaluation LREC, pp. 53-58. Genoa, Italy.
- Pouliquen Bruno, Ralf Steinberger & Camelia Ignat (2003). *Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus*. In: Proceedings of the EUROLAN Workshop Ontologies and Information Extraction at the Summer School The Semantic Web and Language Technology - Its Potential and Practicalities. Bucharest, Romania.
- Pouliquen Bruno, Ralf Steinberger & Clive Best (2007a). *Automatic Detection of Quotations in Multilingual News*. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP. Borovets, Bulgaria.
- Pouliquen Bruno, Ralf Steinberger & Jenya Belyaeva (2007b). *Multilingual multi-document continuously updated social networks*. Proceedings of the RANLP Workshop Multi-source Multilingual Information Extraction and Summarization (MMIES'2007). Borovets, Bulgaria.
- Pouliquen Bruno, Olivier Deguernel & Ralf Steinberger (2008). *Story tracking: linking similar news over time and across languages*. In: Proceedings of the CoLing'2008 workshop: Multi-source, multilingual information extraction and summarization, Manchester, August 2008.
- Pouliquen Bruno, Ralf Steinberger, Camelia Ignat, Emilia Käsper & Irina Temnikova (2004). *Multilingual and cross-lingual news topic tracking*. In: Proceedings of the 20th International Conference on Computational Linguistics (CoLing'2004), Vol. II, pages 959-965.
- Salton G. & C. Buckley (1988). *Term-weighting approaches in automatic text retrieval*. Information Processing & Management, 24 (5): 513-523.
- Schultz J. Michael & Mark Liberman (1999). *Topic detection and Tracking using idf-weighted Cosine Coefficient*. DARPA Broadcast News Workshop Proceedings.
- Sebastiani Fabrizio (1999). *A Tutorial on Automated Text Categorisation*. In: Analia Amandi and Alejandro Zunino (eds.): Proceedings of ASAI-99, 1st Argentinean Symposium on Artificial Intelligence, Buenos Aires, Argentina, pp. 7-35.
- Steinberger Ralf & Bruno Pouliquen (2007). *Cross-lingual Named Entity Recognition*. In: Satoshi Sekine & Elisabete Ranchhod (eds.). *Linguisticæ Investigationes LI 30:1*, pp. 135-162. Special Issue *Named Entities: Recognition, Classification and Use*.
- Steinberger Ralf, Bruno Pouliquen & Camelia Ignat (2004). *Providing cross-lingual information access with knowledge-poor methods*. In: Andrej Brodnik, Matjaž Gams & Ian Munro (eds.): *Informatica. An international Journal of Computing and Informatics*. Vol. 28-4, pp. 415-423. Special Issue 'Information Society in 2004'.
- Steinberger Ralf, Bruno Pouliquen & Camelia Ignat (forthcoming). *Using language-independent rules to achieve high multilinguality in text mining*. In: Françoise Fogelman-Soulié, Domenico Perrotta, Jakub Piskorski & Ralf Steinberger (eds): *Mining Massive Data Sets for Security*. IOS-Press, Amsterdam, Holland.
- Steinberger Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş & Dániel Varga (2006). *The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages*. Proceedings of the 5th International Conference on Language Resources and Evaluation LREC, pp. 2142-2147. Genoa, Italy.
- Tanev Hristo (2007). *Unsupervised Learning of Social Networks from a Multiple-Source News Corpus*. Proceedings of the RANLP Workshop *Multi-source Multilingual Information Extraction and Summarization (MMIES'2007)*. Borovets, Bulgaria.
- Vinokourov Alexei, John Shawe-Taylor & Nello Cristianini (2002). *Inferring a semantic representation of text via cross-language correlation analysis*. Proceedings of Advances of Neural Information Processing Systems 15.
- Vitas Duško, Cvetana Krstev & Denis Maurel (2007). *A note on the Semantic and Morphological Properties of Proper Names in the Prolex Project*. In: Satoshi Sekine & Elisabete Ranchhod (eds.). *Linguisticæ Investigationes LI 30:1*, pp. 115-134. Special Issue *Named Entities: Recognition, Classification and Use*.
- Wactlar Howard (1999). *New Directions in Video Information Extraction and Summarization*. Proceedings of the 10th DELOS Workshop, Sanorini, Greece.
- Wayne Charles (2000). *Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation*. Proceedings of 2nd International Conference on Language Resources and Evaluation (LREC).