Corela
Cognition, représentation, langage

HS-2 | 2005
Le traitement lexicographique des noms propres

Multilingual person name recognition and transliteration

Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Irina Temnikova and Anna Widiger

Electronic version
URL: http://journals.openedition.org/corela/1219
DOI: 10.4000/corela.1219
ISSN: 1638-573X

Publisher
Cercle linguistique du Centre et de l'Ouest - CerLICO

Electronic reference
Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Irina Temnikova and Anna Widiger, "Multilingual person name recognition and transliteration", *Corela* [Online], HS-2 | 2005, Online since 02 December 2005, connection on 02 April 2021. URL: http://journals.openedition.org/corela/1219 ; DOI: https://doi.org/10.4000/corela.1219

Multilingual person name recognition and transliteration

Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Irina Temnikova and Anna Widiger

We thank the whole team of the Web Technology sector at the JRC for providing us with the valuable news data to test the tools, as well as for their technical support. We also want to thank Carlo Ferigato who introduced us to various fuzzy matching techniques. We thank Tomaž Erjavec for helping us with the Slovene language, and Helen Salak for providing us with knowledge about Farsi.

Introduction

1 Many large organisations continuously monitor the media, and especially the news, to stay informed about events of interest, and to find out what the media say about certain persons, organisations, or subjects. Software tools that automatically pre-select the news articles of interest and that pre-process the chosen text collection simplify the daily repetitive task of media monitoring. Crestan & de Loupy (2004) showed that Named Entity extraction and visualisation help users to browse large document collections more quickly and efficiently. This seems plausible as, according to Gey (2000), 30% of content-bearing words in news are proper names.

2 In news analysis it is important to know *What* is the subject, *Who* is being talked about, *Where* and *When* things happened, and *How* it was reported. This paper focuses on the occurrence of proper names in news, i.e. the *Who* part of the analysis. Previous work focused on answering the questions *What* (Pouliquen et al. 2004b), *Where* (Pouliquen et al. 2004a) and *When* (Ignat et al. 2003). Due to the highly multilingual work environment in the European Commission – an organisation with twenty official languages – multilinguality of tools and the cross-lingual aspect are of prime importance.

3 Our analysis is applied to the output of the *Europe Media Monitor* system EMM (Best et al., 2002). EMM is a software toolset that monitors a daily average of 25,000 news

articles in currently 30 languages, deriving from 800 different international news sources. For a subset of about 15,000 articles per day in currently eight languages, we apply unsupervised hierarchical clustering techniques to group related articles separately for each language. We then track related news clusters within the same language and across six of the languages (Pouliquen et al. 2004b)[1]. The JRC's name recognition tools are applied to each of these clusters, i.e. each group of related texts is treated as one meta-text, for which person and geographical place names are extracted and keywords are identified.

4 After giving some background on name transliteration and referring to related work (Section *Background and related work*), we describe tools to identify names in text (Section *Proper name recognition*) and the mechanism to merge name variants, including those written in Cyrillic, Arabic, and Greek script (Section *Detecting and merging name variants*). This is followed by evaluation results (Section *Evaluation*) and by a section on learning relationships between people and how the automatically generated information on names can be used in automatic news analysis (Section *Using names to explore document collections*).

Table 1: Overview of recognised person name in various languages where the various orthographies refer to the same person.

Language	*text*
English	…death of *former Prime Minister* **Rafik Hariri**, blamed by many opposition…
Spanish	…asesinato del *exprimer ministro* **Rafic al-Hariri**, que la oposición atribuyó…
French	…l'assassinat de l'*ex-dirigeant* **Rafic Hariri** et le départ du chef de la …
Dutch	na de moord op *oud-premier* **Rafiq al-Hariri** gingen gisteren bijna een…
German	… *libanesischen Regierungschef* **Rafik Hariri** vor einem Monat wichtige…
Slovene	danjega *libanonskega premiera* **Rafika Haririja**. Libanonska opozicija si…
Estonian	mööndumisele *ekspeaminister* **Rafik al-Hariri** surma põhjustanud…
Arabic	…اغتيال *رئيس الوزراء السابق* **رفيق الحريري** بأياد يهودية وما حدث سابقا …
Russian	…*Бывший премьер-министр Ливана* **Рафик Харири**, который…

THE ITALICS BEING THE RECOGNISED TRIGGER WORD(S).

Background and related work

5 This section gives some background and points to state-of-the-art applications regarding named entity recognition (See *Named entity recognition*), transliteration of person names and their mapping with European name variants (See *Transliteration of proper names*), and the usage of graphs showing relations between persons (See *Relation Maps*).

Named entity recognition

6 Though Named Entity Recognition (NER) is a known research area (e.g. MUC-6 1995, Daille & Morin 2000), multilingual Named Entity Recognition is quite new (ACL-MLNER 2003, Poibeau 2003). Moreover, the cross-lingual aspect (detecting the same names across languages) is often limited to single language pairs or can only be trained on parallel text.

7 People's names can be recognised in text (a) through a lookup procedure if a list of known names exists, (b) by analysing the local context (e.g. 'President' *Name Surname*),

(c) because part of a sequence of candidate words is a known name component (e.g. 'John' Surname), or (d) because the sequence of surrounding parts-of-speech indicates to a tagger that a certain word group is likely to be a name. Sometimes, Machine Learning approaches are used for recognising names within their context by looking at words surrounding known names. For the European languages, it is sufficient to consider only uppercase words. Other languages, such as Arabic, do not distinguish case. At the JRC, we currently use methods (a) to (c), but do not use part-of-speech taggers, because we do not have access to such software for all languages of interest. We currently restrict the recognition to names consisting of least two parts. Until now, the focus has been on people's names, but we also recognise some organisation names.

Transliteration of proper names

8 Transliteration is the process of representing words from one language using the alphabet or writing system of another language (Arbabi et al., 1994). Transliteration is used for formulating concepts mainly existing in one language (e.g. Sharia law) into another, or for reporting about names of people, organisations or places. Transliteration from a language like Arabic would differ depending on the target language. An example is the Arabic name ãÍãÏ, which could be transliterated into English as 'Muhamed' or 'Muhammed', while a likely French transliteration would be 'Mohamed' or 'Mohammed'.[2]

Specificity of transliterating person names

9 Many publications, web sites and transliteration schemes exist for languages that use the Cyrillic, Greek, or Arabic alphabets, but most of them apply to general words rather than to person names. The fundamental difference between transliterating natural language words and transliterating names is that the pronunciation of words normally follows some conventions, meaning that hand-crafted linguistic equivalence rules can be used. While the same may be partially true for names of the same language (e.g. Russian names in Russian text), transliteration becomes more difficult when the names found are of international origin – as it is often the case in news articles. For instance, in a Russian news article it is likely that names of French, Italian, English or Arabic origin are found. In order to transliterate such international names efficiently, it would be necessary to know the source of the name as this tells us about the target language equivalence. If the origin of the name *Chirac*, for example, is known as being French, then it is pronounced as /ʃi1ak/ and should be transcribed as ÔíÑÇß in Arabic, or ШИРАК in Russian. However, if it were an Italian name, it would be pronounced as /kirak/ and transliterated as ßíÑÇß in Arabic and КИРАК in Russian.

Dealing with many language-pairs

10 Because of the language-dependence of transliteration, previous work in automatic name transliteration has always been carried out for specific language pairs such as Chinese-English or Russian-English, as can be seen in the large enumeration of previous work in Lee et al. (2005). Although it is likely that this limitation to specific language pairs produces better results than our more language-independent approach, such language-dependent approaches are not a useful option in the context of our highly

multilingual news analysis system, which aims at dealing with twenty or more languages and where the original language of names is usually not known.

Transliteration challenges

11 The transliteration of names from each writing system poses its own challenge. The Cyrillic and Greek scripts seem to be most similar to the Latin script in that they are basically phonetic: letters or groups of letters correspond to specific sounds. The major problems are (a) phoneme-letter equivalences are in an *n-to-n* relationship (i.e. a letter can often be pronounced in different ways and a certain sound can be written with different letters), and (b) the phoneme inventory in different languages (and writing systems) differs: If a language does not know a sound, it will transliterate this sound by another similar one. When back-transliterating the name, the spelling is thus likely to be wrong. For instance, the German and English sound for the letter 'h' is unknown in Russian and is frequently transliterated into 'г', pronounced /g/. Examples are the city name Heidelberg (гейдельберг', pronounced /gejdɛljberk/) and Harry Potter (гарри поттер, pronounced /garipotɛr/). When these names are found in Russian text and are back-transliterated into English or German, they will thus appear as 'Geidelberg' and 'Gari Potter', or similar.

Specific challenges for Arabic transliteration

12 Arabic does not have the sounds /p/, /v/ and /g/. 'Paul' is transcribed as بول /bol/, 'Valery' as فاليري (/faliry/), and 'Globe' as غلوب (/1lo:b/). A name such as 'Vladimir Putin' will therefore be transliterated as فلاديمير بوتين (/fladimi:r buti:n/).

13 Transliteration from Arabic to languages using the Latin alphabet (*romanisation*) is additionally made difficult by the fact that short vowels are usually not written in Arabic. Any romanisation effort therefore typically includes *vowelisation*, i.e. the insertion of the short vowels in the target language (Arbabi, 1994). As Arabic dialects differ in pronunciation, vowelisation is clearly dependent on the dialect. This is presumably the reason why, for the unique spelling of the Arabic name ÓáíãÇä, forty different transliterations can be found, including 'Salayman', 'Seleiman', 'Solomon', 'Suleiman' and 'Sylayman'.

Challenges for languages using ideographs

14 Transliteration into languages with an ideographic writing system such as Chinese, where each symbol is equivalent to a concept rather than to a sound, has to be tackled in an entirely different way. Chinese has a system of syllables called *Pinyin* (Swofford 2005), a combination of initial and final sounds which can be used to construct about 300 syllables. When transliterating non-Chinese names, a closest syllable-to-syllable approximation is looked up, and for each syllable a Chinese corresponding ideogram can be chosen from the list of different tone variants. The transcription of an English or German name will thus consist of a concatenation of Chinese syllables. For example, 'Beethoven' would be represented in Pinyin as 'bej-do-fen'.

Methods for transliterating

15 Existing automatic name transliteration systems either use hand-crafted linguistic rules, or they use Machine Learning methods (e.g. Lee et al. 2005), or a combination of both. Arbabi et al. (1994), for instance, use linguistic rules and neural networks to vowelise and romanise Arabic names, as well as to filter out unlikely over-generated target word forms. Lee et al. (2005) learn name transliteration from large bilingual Chinese-English lists of proper names, using the Expectation Maximisation algorithm. They do not use pronunciation dictionaries or manually generated phonetic similarity scores. At the JRC, we are using hand-crafted transliteration rules. The output is then processed by further hand-crafted substitution rules in order to produce an *internal standard representation* (see Section *Detecting and merging name variants*).

Relation maps

16 When a tool extracts person names from documents, it implicitly generates useful information regarding the co-occurrence of persons. Ben-Dov et al. (2004), who worked on both detecting relationships and visualising them, quote: 'knowledge can be created by drawing inference from what is already known' (Davies 1989). Such knowledge or information can be visualised with relation maps.

17 In principle, two methods can be used to generate relation information: (a) the observation of the co-occurrence of names in the same text, and (b) the usage of syntactic-semantic rules to detect more specific relationships between persons. If two persons are often mentioned in the same document (co-occurrence information), they are likely to be in a certain relationship. This relationship is difficult to label, as it could be friendship, rivalry, family relationship, belonging to the same organisation, participation in the same meeting, etc. A rule-based system, on the other hand, would be able to detect more specific relationships. Ben-Dov et al. (2004) compare both approaches and come to the conclusion that, when searching for information about joint meetings, co-occurrence-based algorithms exhibit a good *Recall*, but are bad for *Precision*, while the inverse is true for rule-based methods. The authors estimate that writing rules to identify 'participation in a common meeting' takes a programmer between one and three weeks for one language only, assuming that an appropriate parser is available. The advantage of the co-occurrence-based approach, used by the JRC, is that no rules need to be written and that the same mathematical formulae can be used to describe (co-occurrence) relationships in all languages.

18 The commercial system *Connivence Maps*, by *Connivence*s, presents relationships among actors in the news, but they provide no details about the algorithms used (see http://www.connivences.info/ last visited 06/06/2005).

Proper name recognition

19 At the JRC, we add all names detected during our daily news analysis to a database of known names, so that these names can then be recognised in the future by a simple lookup procedure (method (a) described in Section *Named entity recognition*). After one year of news analysis, the database has grown to about 150,000 distinct names (not counting variants of the same name; see Section *Detecting and merging name variants*).

More than 500 new names are inserted every day. For performance reasons, a Unicode (UTF-8) compatible Finite State Automaton is used. A set of regular expressions is generated for each entry of the database as input to the FLEX utility (Paxson 1995), which generates the automaton. In order to exclude the recognition of name variants due to typing errors, the automaton only searches for names that were found at least twice. To date, the tool thus searches about 50,000 persons, representing about 60,000 different orthographies.

Trigger words

20 To guess new names (method (b) described in Section *Named entity recognition*), an extensive list of local patterns was developed in a boot-strapping procedure: We first wrote simple local patterns in PERL to recognise names in a collection of three months of English, French and German news. We then looked at the most frequent left and right hand side contexts of the resulting list of known names. For English alone, we currently have about 1,100 local patterns, consisting of titles ('Dr.', 'Mr', etc.), country adjectives (such as 'Estonian'), professions ('actor', 'tennis player', etc.), specific patterns (such as '[0-9]+ year-old'), etc. We refer to these local patterns as *trigger words*. For each added language, native speakers translate the existing pattern lists and use the same bootstrapping procedure to complete the patterns.

21 Those patterns allow the program to recognise new names (i.e. in 'the American doctor John Smith'), but a stored list of such patterns is also useful to give users additional information about persons. In the previous example, for instance, the user will see that *John Smith* probably is an American doctor. When a name is often used with the same trigger words, statistical measures can be used to qualify names automatically. For instance, *George W. Bush* will be recognised as being the American president, *Rafik Hariri* as being the 'former Lebanese prime Minister', etc.

22 Currently the JRC has rules for the following languages: English, French, German, Spanish and Italian. To a certain extent we have also some Dutch, Estonian and Slovene patterns. A first version of Russian is almost ready, Arabic is under development. The aim is to include all twenty official languages of the European Union and candidate countries.

Table 2: two examples of patterns used to recognize *Tony Blair* and *Romano Prodi* in Slovene texts

Tony(a|o|u|om|em|m|ja|ju|jem)?\s+Blair(a|o|u|om|em|m|ju|jem|ja)
Roman(a|o|u|om|em|m|ju|jem|ja)?\s+Prodi(a|o|u|om|em|m|ju|jem|ja)?

Dealing with declension

23 In some languages, especially in Slavonic and Finno-Ugric languages, both the local patterns and the proper names are inflected and can have suffixes, as can be seen in the Slovene example 'Tožba proti Donaldu Rumsfeldu zaradi mučenj'. The automaton to recognise names therefore has to allow for a variety of suffixes (in the given example, the suffix 'u' was added to the name *Donald Rumsfeld*). Some of the hand-written rules

used at the JRC to detect person and place names simply consist of possible suffix lists for each name. Others are more complex, for instance using substitution functions to detect the Estonian *New Yorgile* as an inflection of *New York* or detecting that the 'o' in *Romano Prodi* is part of the name and should not be identified as the 'o'-suffix in Slovene text. Table 2 shows two sample suffix lists that are required to detect known names in Slovene text. Table 3 lists the rules used to generate automatically inflected variants for Russian names in our database.

Table 3: Simplified rules to build a pattern recognising all possible declensions of a given name in Russian text.

Ending	Pattern	Example
-а	а|ы|и|е|у|ой	Никита Никиты Никите Никиту Никитой *Nikita Nikity Nikite Nikitoy Nikitu* Ольга Ольги Ольге Ольгу Ольгой *Oljga Oljgi Oljge Oljgu Oljgoy*
-я	я|и|ю|е|ей|ёй|ю	Илья (*Iljya*) Ильи Илье Илью Ильёй Дарья (*Darya*) Дарьи Дарье Дарью Дарьей
-ь	ь|и|ью|я|ю|ем|е	Любовь (*Lyubovj*) Любови Любовью Игорь (*Igorj*) Игоря Игорю Игорем Игоре
-й	й|я|ю|ем|е|и	Андрей (*Andrey*) Андрея Андрею Андреем Андрее Анатолий (*Anatoliy*) Анатолия Анатолию Анатолием Анатолии
-ел	ел|ла|лу|лом|ле	Павел (*Pavel*) Павла Павлу Павлом Павле
-ев	ев|ьва|ьву|ьвом|ьве	Лев (*Lev* translated as 'Leo') Льва Льву Львом Льве
-о -у -е -э -и	- (not declined)	Марко (*Marko*) Мари (*Marí*) - *French 'Marie'* Андрэ (*Andre*)
Default case: consonants	а|у|ом|ем|е	Иван (*Ivan*) Ивана Ивану Иваном Иване Джордж (*Dzhordzh*) Джорджа Джорджу Джорджем Джордже *English. 'George'*

Storage of names in a database

24 Names identified in any of the analysed languages are automatically stored in a database, together with information on where and when the name was found and with information on the language of the text. The trigger words found around the name are also stored. Each distinct name is assigned a numerical identifier. Variants identified for the same name (see Section *Detecting and merging name variants*) are all stored with the same identifier. To add additional name variants, especially in non-European languages, we automatically search the free Wikipedia[3] online encyclopaedia for all names in our database (*Cf. Figure*). When a Wikipedia entry exists, we add the corresponding URLs to the database to allow users to find additional information about a certain person. Additionally we copy the photograph of the person, when available.

25 When we detect new names, we use a fuzzy matching tool to automatically detect whether the name is a variant of a name already present in the database (see Section *Fuzzy matching of name variants*).

Table 1 demonstrates how difficult the name recognition can be across languages.

Figure 1: Entry for Rafik Hariri in the Wikipedia encyclopaedia (http://en.wikipedia.org/wiki/Rafik_Hariri), and some name variants detected automatically

26 Entry for Rafik Hariri in the Wikipedia encyclopaedia (http://en.wikipedia.org/wiki/Rafik_Hariri), and some name variants detected automatically

Detecting and merging name variants

27 For many person names, several variants are used in the media, not only across languages (see Table 1), but often even within the same language (in 50 English articles published on the 14th April 2005, we found four orthographies: *Rafik Hariri*, *Rafik al-Hariri*, *Rafiq Hariri* and *Rafiq al-Hariri*). In order to allow users find information about certain persons independently of the name spelling, we aim at storing all name variants together with one unique numerical identifier.

28 Using the similarity of name orthography (described in Section *Fuzzy matching of name variants*), we currently merge name variant candidates automatically if they are found in the same news cluster and if their similarity score is high enough (70%). As clusters can consist of between 2 and 100 articles talking about the same event (for details see Pouliquen et al. 2004b), it is quite likely that two variants of the same person name are found in the same cluster.

29 As the system to match names *across* languages is still under development, cross-lingual name variant merging is currently done only if two variants are extremely close (i.e. similarity more than 95%). When a new name is detected, its similarity with all other names is computed. Then the process automatically merges similar names (see Table 4 for examples compiled for one day). For the others (similarity between 80% and 95%), the system displays a list of new names similar to previous ones (variant candidates), asking for a human confirmation before merging them. As shown in the examples in

Table 5, all names for that day need to be merged. Even the case of *Daniel Cicarelli* turned out to be a typo so that the two names should be merged[4].

30 As we do not currently consider the context of names, it could happen that the system merges names such as 'Mariana Gonzalez' (a Venezuelan fencer) and 'Mariano Gonzalez' (an Argentinean football player). The system therefore allows manual intervention to correct incorrectly merged names or to merge two variants that have not been detected automatically.

31 As shown in Table 4, Table 5 and Footnote 4, quite a few misspelled names appear in the news, but it is important to capture them anyway in order to improve the *Recall*.

Table 4: List of extremely similar names found in the news of a single day (30 May 2005).

New name	Merged with existing name:
Abdüllatif Sener	Abdullatif Sener
Abubakar Tanko	Aboubakar Tanko
Allan McDonald	Alan McDonald
Bahiya al-Hariri	Bahia al-Hariri
Brian Herta	Bryan Herta
Eid Cabalu	Eid Kabalu
Hassan Mohamed Nur	Hassan Mohamed Nuur
Ismail al-Hadithi	Ismail al Hadithi
Johana Melka	Johanna Melka
José Luis Lingeri	Jose Luis Lingeri
Luis Fernández	Luis Fernandez
Michael Haefrati	Michael Haephrati
Mohamed Dhia	Mohammed Dhiaa
Nikolas Sarkozy	Nicolas Sarkozy
Salomé Zurabishvili	Salome Zurabishvili
Sergei Brin	Sergey Brin
Stanley Fisher	Stanley Fischer
Surat Ikramov	Sourat Ikramov
Trudi Stevenson	Trudy Stevenson

Werner Schneyder	Werner Schneider

THESE VARIANTS ARE AUTOMATICALLY MERGED.

Table 5: List of rather similar new names (30 May 2005).

Russ Young	Ross Young
Gary Shafer	Gary Sheffer
Mohammed Dhia	Mohammad Dhiya
Brian Vilora	Brian Viloria
Saad al-Harir	Saad al-Hariri
Pierre Gadonnaix	Pierre Gadonneix
Abudullahi Yusuf	Abdullahi Yusuf
... (altogether 24 propositions) ...	
Daniel Cicarelli	Daniella Cicarelli

BEFORE MERGING THESE VARIANTS, MANUAL CONFIRMATION IS REQUIRED.

32 Due to the usage of different scripts in Greek, Russian and Arabic, the merging of names in these languages partially differs from the process used for languages written with the Roman alphabet.

Normalisation of the name orthography

33 Name variants across languages are often due to the omission of diacritics. For example a British newspaper can sometimes refer to *François Mitterrand* as *Francois Mitterrand*. A number of further regular variations we observed are the singling of double consonants, transcriptions of *f* by *ph* (e.g. *Ralph Schumacher*), alternative usage of *w* or *v* in Russian names (e.g. *Wladimir* vs. *Vladimir*), alternative spellings of the sound 'u' as *u* or *ou*, etc. In languages such as Lithuanian, transcriptions are common (e.g. *Buš* for *Bush*). We therefore decided to develop an *internal standard representation*, ISR, which has the pragmatic aim of linking the variants, without wanting to make theoretical claims of any sort.

Before calculating a similarity between pairs of names, all names are *standardised* using a set of approximately 30 substitution rules. Examples are:

- accented character → non-accented equivalent
- double consonant → single consonant
- ou → u
- wl (beginning of name)→ vl
- ow, ew (end of name) → ov, ev
- ck → k

- ph → f
- ž → j
- š → sh

34 This list of substitution rules can also contain the most frequent exceptions not covered by the generic rules (e.g.: джеймс => 'James' to avoid the basic transliteration as 'geyms'). Examples of names after this standardisation are:

- Jacques Chirac → Jak Shirak
- Wladimir Ustinow → Vladimir Ustinov
- Vladimir Oustinov → Vladimir Ustinov
- Abdalah Džburi → Abdalah Djburi
- Abdallah Joubouri → Abdalah Juburi
- Malik Saïdoullaïev → Malik Saidulaiev
- Malik Saidullajew → Malik Saidulajev

Transliteration of non-Latin scripts

35 For Greek, Russian and Arabic, which do not use the Latin script, we use hand-written transliteration and adaptation rules to represent names with the Latin alphabet. Transliteration consists of a number of substitution rules that replace one or more non-Latin characters by one or more Latin characters. For Greek, for instance, the following substitutions apply:

- λ → l
- θ → th
- μπ → b

36 After the transliteration, the normalisation rules described in previous Section *Normalisation of the name orthography* are applied. The results of the transliteration and standardisation are often phonetic (e.g. 'Bil Klinton', 'Jak Shirak', etc.), but they are similar enough to the standard representation to produce good results in the fuzzy matching process (see Section Fuzzy matching of name variants). Example results for Greek, Cyrillic and Arabic transformations are:

- Κόφι Ανάν (Greek) → Kofi Anan
- кофи аннан (Russian) → Kofi Anan
- кофи анан (Bulgarian) → Kofi Anan
- ßæÝí ÚäÇä (Arabic) → Kufi Anan
- कोफी अन्नान (Hindi) → Kofi Anan

37 At the JRC, we have developed transliteration rules for the following writing systems: Greek, Cyrillic (Russian, Bulgarian and Ukrainian), Arabic (including Farsi and Urdu) and Devanagari (Hindi and Nepali). Writing the rules to transliterate the Devanagari script took about 2 hours.

Fuzzy matching of name variants

38 In order to identify potential name variants (like those in Table 5) we carry out a pair-wise comparison of all transliterated and standardised names. If the similarity of the pair of names is above a certain threshold, the names are variant candidates.

Figure 2

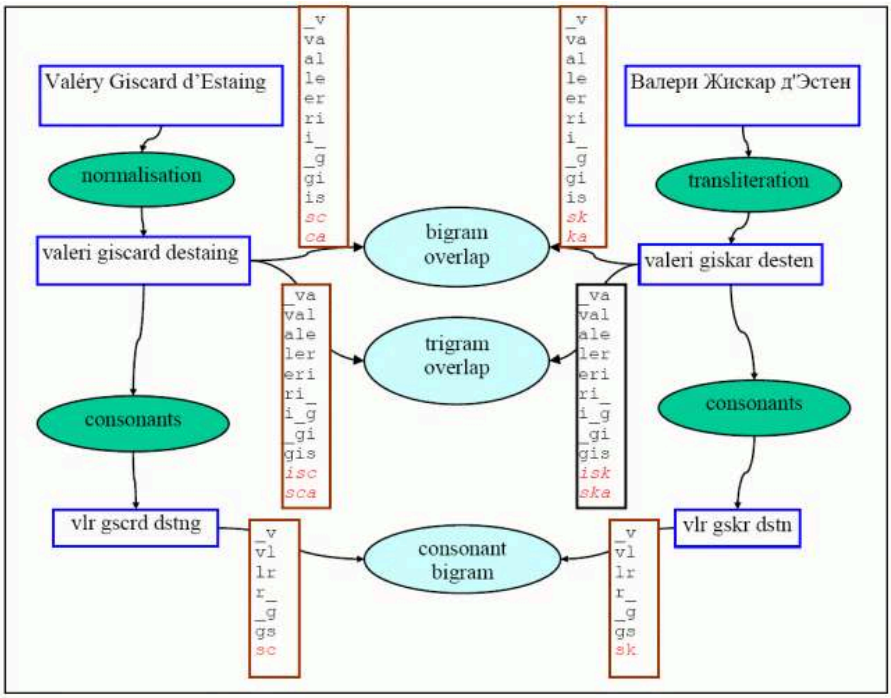

Figure 2: Example of the matching process between a Cyrillic (Russian) and a Latin (French) name

EXAMPLE OF THE MATCHING PROCESS BETWEEN A CYRILLIC (RUSSIAN) AND A LATIN (FRENCH) NAME.

39 For the similarity calculation we combine three similarity measures. We currently take the average of the three measures, but we plan to evaluate the relative impact of each of them in order to optimise their relative weight for the similarity calculation.

40 The three measures are based on letter ngram similarity: we calculate the *cosine* of the letter ngram frequency lists for both names, separately for bigrams and for trigrams. The third measure is the *cosine* of bigrams based on strings without vowels. We do not use phonetic transcriptions of names as these are reported to be less useful than *string-like* approaches (Zobel & Dart, 1995). Moreover, phonetic transcription rules are different from language to language (e.g. *Chirac* would in Italian be pronounced as / kirak/) and finding the transcription rules for many languages would be difficult.

41 Figure 2 gives an overview of the process for comparing a French name with its Russian counterpart written with Cyrillic letters.

Special variation to deal with Arabic

42 Standard Arabic writes long vowels and often omits short ones. When comparing names written in Arabic with names written with the Latin alphabet, we therefore delete vowels from the latter before computing the similarity. For instance, the internal standard representation for the name *Condoleezza Rice* is 'kondoleza rice'. The same name written in Arabic is ßæäÏæáíÒÇ ÑÇíÓ. The result of the transliteration and standardisation of the Arabic version of the name is 'konduliza rais'. The *cosine* of bigrams between these two representations without vowels ('kndlz Rc' and 'kndlz Rs') is rather high (0.875) so that the two names written with the Arabic and the Latin scripts are successfully identified as name variants.

43 Figure 3 summarises the matching process for an Arabic name.

Figure 3

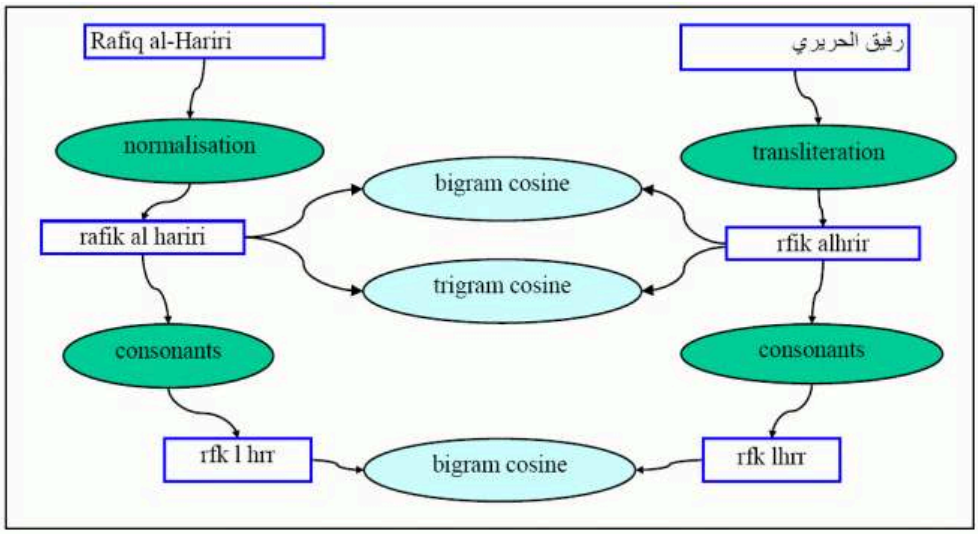

Figure 3: Arabic/Latin name matching example

ARABIC/LATIN NAME MATCHING EXAMPLE

Evaluation

Evaluation of name recognition

44 Our focus is not on optimising Named Entity Recognition for one language, but rather on finding an approach that is easily and quickly adapted to new languages. We nevertheless launched an evaluation on the performance of the tool for various languages:

45 In each language we chose a random selection of about 100 newspaper articles. We applied our person name recognition tool. Experts listed all person names that were present in the text. For each article we then compared if each of the automatically recognised person names was also selected by the expert (to get *Precision*), and if all the manually extracted names were also automatically found (to get *Recall*). We combine those two values using the F-measure[5].

46 We have to emphasise that, unlike in traditional name recognition evaluation, our aim was to identify the presence or non-presence of a name in the text, and that it was not our concern to identify each and every mention of the name. Furthermore we restricted our evaluation to the recognition of person names, ignoring organisations and toponyms. The results are summarised in Table 6.

Table 6: Evaluation of person name recognition in various languages.

Language	# rules	# texts	# names	Average Precision	Average Recall	Average F-measure

English	1100	100	405	92	84	88
French	1050	103	329	96	95	95
German	2400	100	327	90	96	93
Spanish	580	94	274	85	84	84
Italian	440	100	298	92	90	91
Russian	447	61	157	81	69	74

THE NUMBER OF RULES (I.E. TRIGGER WORDS) GIVES AN IDEA OF THE EXPECTED COVERAGE FOR THIS LANGUAGE. THE THIRD AND FOURTH COLUMNS SHOW THE SIZE OF THE TEST SET (NUMBER OF TEXTS, NUMBER OF MANUALLY IDENTIFIED PERSON NAMES).

47 The results are less good than for named entity recognition systems that use part-of-speech taggers, are optimised for a given language, and do not aim at such high multilinguality. Our *Precision* is nevertheless reasonably high. In our setting, where we try to detect names in *clusters* of news instead of in individual articles, the lower *Recall* is not a big problem, because names are usually found in at least one of the articles so that the person information for the cluster is often complete.

48 The low Recall score could be due to the nature of our heterogeneous test set: The set not only includes articles from many different domains (politics, sports results, discussions of television programmes, etc.), but also from international newspapers from all over the world (especially for the English language texts).

49 The system has to analyse articles such as: 'Phe Naimahawan, of Chiang Mai's Mae Ai district, has been selected (...) to represent Thailand in a swimming event (...). Phe is being helped by Wanthanee Rungruangspakul, a law lecturer'. Without part-of-speech tagging, it is difficult to guess that 'Phe Naimahawan' is a person name. However, in the same text, we were able to guess the name 'Wanthanee Rungruangspakul' thanks to the trigger word 'law lecturer'.

50 The lower *Precision* for German was predictable as in German every noun is uppercased, which often results in the system recognising common nouns as proper names. In the example: "Die österreichische Eishockey Nationalmannschaft bekommt während der Heim-WM noch Verstärkung", 'Eishockey Nationalmannschaft' (*Ice hockey national team*) is wrongly triggered by 'österreichische' (*Austrian*).

51 The relatively bad scores for Spanish are due to various facts. One of them was that we did not have any Basque first names in our name lists and that many Basque names were found in the test set. Another reason was that our system frequently only recognised the first part of the typical Spanish compound names. Finally, several organisation names were classified by the algorithm as person names.

52 The explanation for the lower Russian results mainly is that our name database contained only a dozen Russian names so that the system had to guess most names, which is harder than looking up known names.

Evaluation of transliteration

53 An unbiased evaluation of the variant matching algorithm for names written with the Latin script is not possible because all frequent variants are already stored in the database, and some of them had already been manually checked or were added via the Wikipedia search (see Section *Storage of names in a database*). We would only be able to test the system on new names, but for these we would not find variants in the database. Testing the system on previously unseen variants is not particularly useful either. Instead, we evaluated how accurately the system identified the Latin equivalent of names written with Cyrillic (Russian) and Arabic letters. For this purpose, two native-speakers prepared a short list of randomly selected names that they found in the news of the day. We then verified whether or not the system proposed the European version of this name as the most similar (with a minimum threshold of 50%). We must highlight that each of the names was compared to the orthographies of 150,000 other persons.

54 This test allows us to see whether the transliteration, standardisation and fuzzy matching tool works properly. Moreover, it allows us to validate whether our database contains the most important names.

Figure 4

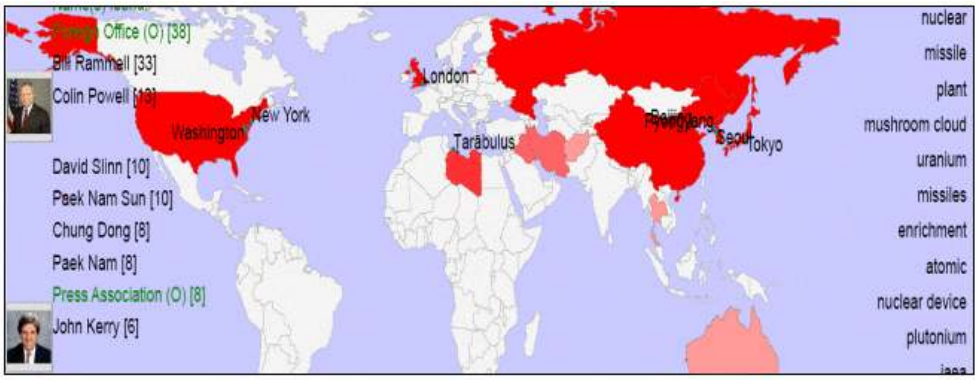

Figure4: Interactive visualisation of places, names and terms found in news clusters

INTERACTIVE VISUALISATION OF PLACES, NAMES AND TERMS FOUND IN NEWS CLUSTERS.

Russian name variant identification

55 Out of 53 names analysed, only one did not have a Latin equivalent in the database, but the system still returned a wrong proposal. In two other cases, the wrong person was identified. The remaining 50 names were successfully matched to the correct person. Both *Precision* and *Recall* were thus 94.3%.

56 The person not present in our database was *Robert Bradtke* (the American Secretary for European and Eurasian Affairs). This name was wrongly matched with *Robert Bartko* (a German cyclist).

57 The two false hits were *Jean-Claude Juncker* (transliterated as 'Jan-Klod Yunker' and matched with *Carsten Jancker*), and *Heinz Fischer* (transliterated as 'Khaynts Fisher' and matched with *Joschka Fischer*).

Arabic name variant identification

58 All of the 30 names selected had a Latin-script equivalent in the database. However, two names were not found and three names were assigned to the wrong person. The remaining 25 names were matched successfully. *Precision* is thus 89.3% and *Recall* is 83.3%.

59 Among the good examples, *Jean-Pierre Raffarin*, transliterated as 'Jan-Biar Rafaran', was still matched; and similarly *Arnold Schwarzenegger*, transliterated as 'Arnuld Shuarznijr'. Even short names such as *Jack Straw*, transliterated as 'Jak Stru', were found.

60 The two names not found were due to bad transliteration: *John Garang* has the Arabic variant Ìæä ÞÑäÞ which was transliterated as 'Jon Qrnq' and was not similar to any names in the database. The same is true for ÌæÑÌ ßáæäí, which was transliterated as 'Jurj Kloni' and should have been identified as *George Clooney*.

61 Wrongly matched names were *John McCain*, transliterated as 'Jon Mak Kin' (and matched with *Jean Makoun*), *Colin Powell* transliterated as 'Kuln Baul', and *Michael Jackson* as 'Maikl Jakson'. An obvious solution would be to manually add transliteration rules for the most common names (George, John, Michael, etc.).

Farsi name variant identification

62 22 names (found from online articles on BBC world service[6]) were selected. All of them were actually in our database, 20 were found as being the most similar, but the system did not find two names (ÎÇãäå Çی, *Ali Khamenei* and ãÍãÏ ÓÚیÏ , *Mohammad Saeedi*).

Using names to explore document collections

63 The tools to recognise and match names are part of a larger system to analyse multilingual document collections, by grouping related documents, extracting information from them and visualising some of

Figure 5

Figure 5: NewsExplorer entry for Rafiq Hariri

NEWSEXPLORER ENTRY FOR RAFIQ HARIRI

64 the results. A major purpose of the system is to allow users to sieve through large amounts of documents quickly. The following sections show applications where names detected automatically from multilingual news collections are used.

Visualisation of names on geographical maps

65 For each cluster of related news articles detected by the *Europe Media Monitor* system (EMM), we extract place names and generate an interactive map showing the geographical coverage of the articles (Pouliquen et al., 2004a and 2004b; see *Cf.* Figure 4). Additionally, a number of keywords identified for the cluster and the names detected in this cluster are listed on the map. For each cluster of related news articles, users can thus see various information aspects at a glance. In a customised version of the tool, users can also see on the same map which of their manually selected search terms were found. The map is generated using *Scalable Vector Graphics* (SVG) and is interactive so that users can zoom into a specific geographic area. The interactive feature allows them furthermore to see the context in which place names, persons, and terms were mentioned, and hyperlinks allow to jump to specific text passages. This visualisation tool even allows users to get an overview of the contents of text collections written in languages they may not understand.

Name browser

66 In the JRC's News Explorer system, the information collected during the daily multilingual news analysis is stored in a relational database so that information about

past events, persons and places can be browsed. For each cluster, in currently eight languages, the system keeps track which people are mentioned together with which other people, countries, and keywords. As the database is updated every day, a network of links builds up over time. For instance, the database can be queried for all news clusters that mention a certain person, and it can tell which other persons were mentioned in the same clusters. For each news cluster, a link to the original URL of the most typical article (the *medoid*, the one closest to the cluster centroid) allows the users to read up on the story.

A web interface gives access to the information stored about each person. This information includes:

- information about the person itself: name, name variants, photograph (when available);
- clusters this person was mentioned in;
- the trigger words (*titles*) most frequently identified for the clusters associated with this person;
- a list of *associated* persons, i.e. those person that are frequently mentioned in the same news clusters.

67 Additionally, a daily *VIP list* displays the persons most often mentioned in the news of that day.

68 As the titles are stored in the database, the user can also query all persons having the *title* 'Georgian president', and similar. For details on the browsing functionalities, see Steinberger et al. (2005).

69 Most of the information is exported to a public web site (http://press.jrc.it/NewsExplorer/), as shown on Figure 5.

Identifying links between persons

70 When displaying the associated persons ranked by frequency, the people that are in the news all the time (e.g. *George Bush*) will appear in almost all the lists. We therefore introduced a weighting factor that allows to down-weight highly frequent names and to focus on those person names that are specifically associated with a given other person. The weighting formula uses three factors: the number of clusters each person appears in, the number of common clusters two persons appear in, and the number of 'further associates' each of the persons have. The formula computes a *specific association weight* between two entities in our database:

Equation n°1. Relationship weight between two entities

$$w_{e_1,e_2} = Co_{e_1,e_2} \cdot Icf_{e_1,e_2} \cdot Iass_{e_1,e_2}$$

71 Where:

e_i: Entity

$Co_{e1,e2}$: Cluster co-occurrence between e_1 and e_2

$Icf_{e1,e2}$: Inverse cluster frequency of e_1 and e_2

$Iass_{e1,e2}$: Inverse association frequency of e_1 and e_2

Equation n°2. Cluster co-occurrence weight

$$Co_{e_1,e_2} = 1 + \ln(C_{e_1,e_2})$$

Where:
$C_{e1,e2}$: Number of clusters where e1 and e2 occurring together

Equationn°3. Inverse cluster frequency

$$Icf_{e_1,e_2} = \frac{2C_{e_1,e_2}}{(C_{e_1} + C_{e_2})}$$

Where:
$C_{e1,e2}$: Number of clusters where e1 and e2 appear together
C_{ei}: Total number of clusters where e_i appears; i=1,2

Equation n°4. Inverse association frequency

$$Iass_{e_1,e_2} = \frac{1}{1 + \ln(A_{e_1} . A_{e_2})}$$

Where:
A_{ei}: Total number of entities occurring with e_i ; i=1,2

72 The weighted list of associated persons shows rather different names from the pure frequency list. For the Secretary-General of the Council of the European Union *Javier Solana*, for instance, the most frequently co-occurring names are the well-known politicians *George Bush*, *Jacques Chirac*, *Yasser Arafat* and *Kofi Annan*. In the weighted list, however, the two top-ranking persons are *Christina Gallach* (Solana's spokesperson) and *Pierre de Boissieu* (Solana's assistant). These two persons are less widely known because they are not mentioned much outside the context of Javier Solana, but their names are very closely linked to Solana as they are typically mentioned in the news when Solana is mentioned.

Displaying relation maps

Figure 6

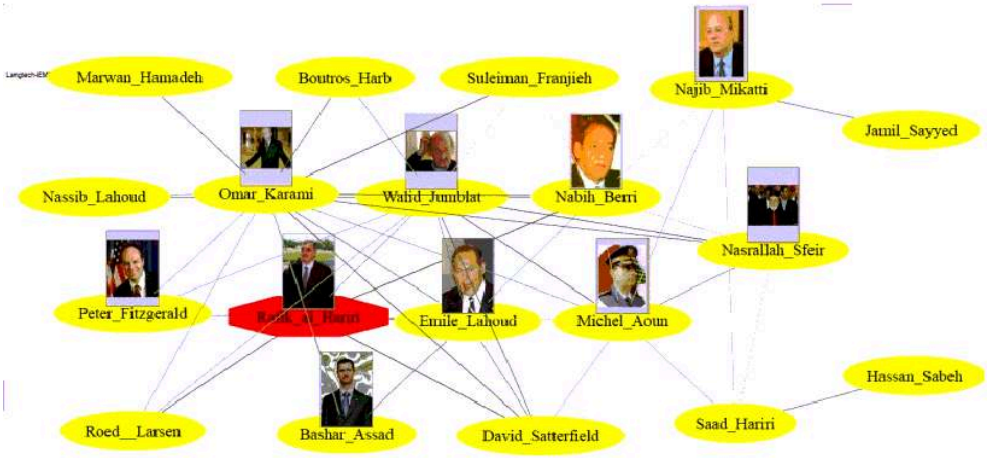

Figure 6: Relation map showing Rafik Hariri and the 20 persons most related to him (N=20)

RELATION MAP SHOWING RAFIK HARIRI AND THE 20 PERSONS MOST RELATED TO HIM (*N*=20)

73 Once we are able to weight relationships between persons, we can summarise the links among persons using a 'relation map' (see Section *Relation maps*). We use the *graphviz* open source graph visualisation software[7], and more specifically the *neato* utility based on the algorithm of Kamada & Kawai (1989), which uses a 2-D graph to display the closest nodes together. For a given person *A*, we give as input a non-oriented graph where each node is a person and each edge is a co-occurrence relationship (using the weight described in previous Section *Identifying links between persons*). A graph takes as a parameter a person and computes the undirected graph. A filter then allows to display only the first *N* relations (those *N* relations having the highest weight). In Figure 5, the user can get an overview of the persons occurring together with *Rafik Hariri*. The user can then select another person and display his or her corresponding graph. This graph is useful to give a quick overview of various groups of persons related to this person *A*.

Conclusion and future work

74 Many of the tools mentioned in this paper are already in daily use, but others still have to mature and stabilise. The cross-lingual matching of name variants already produces useful results for an interactive system, but the merging of name variants cannot yet be fully automated because it still produces errors. We would like to explore how the cluster context of two names can be used to improve the quality of the name merging tool. Comparison of time series like in Shinyama & Sekine (2004) could improve the *Precision*.

75 We also plan to dedicate more time to improve the name recognition itself. Some patterns could recognise organisation names. We would like to explore systems to automatically (or semi-automatically) extend patterns to new languages.

76 Currently we use the content of the Wikipedia entries only to get cross-lingual links and the photograph picture of the person. Interesting research would be to mine these Wikipedia texts automatically for further information. The relationship between

people, for instance, could be confirmed if a given person is mentioned in somebody else's page.

BIBLIOGRAPHY

Arbabi M., Fischthal S.M., Cheng V.C. & Bart E. (1994). Algorithms for Arabic name transliteration. IBM J. Res. Develop. Vol. 38, No. 2, March 1994

Ben-Dov Moty, Wu Wendy, Feldman Ronen & Cairns Paul (2004), Improving knowledge discovery by combining text-mining and link analysis techniques. SIAM International. Conference. on Data Mining, Florida, USA.

Best Clive, van der Goot Erik, de Paola Monica, Garcia Teofilo & Horby David (2002). Europe Media Monitor – EMM. JRC Technical Note No. I.02.88. http://emm.jrc.cec.eu.int

Crestan Eric, de Loupy Claude, (2004), Browsing Help for a Faster Retrieval, In: Coling2004 proceedings, pp. 576-582. Geneva, August 2004

Daille Béatrice, Morin Emmanuel, (2000) Reconnaissance automatique des noms propres de la langue écrite: les récentes réalisations, Traitement Automatique des Langues (TAL), Vol. 41(3), pp.601-622, 2000.

Davies Roy (1989), The creation of new knowledge by information retrieval and classification. The Journal of Documentation, Vol. 45, No 4, pp. 273 –301, December 1989

Gey Frederic (2000). Research to Improve Cross-Language Retrieval – Position Paper for CLEF. In C. Peters (ed.): Cross-Language Information, Retrieval and Evaluation workshop. CLEF 2000, pp. 83-88, Lisbon, September 2000

Ignat Camelia, Pouliquen Bruno, Ribeiro António & Steinberger Ralf (2003). Extending an Information Extraction Tool Set to Central and Eastern European Languages. RANLP 2003 proceedings, pp. 33-39. Borovets, Bulgaria, September 2003.

Kamada T. and Kawai S. (1989) An algorithm for drawing general undirected graphs. Information Processing Letters, 31(1):7–15, April 1989.

Lee Chun-Jen, Chang Jason S. and Jang Jyh-Shing Roger (2005), "Extraction of Transliteration Pairs from Parallel Corpora Using a Statistical Transliteration Model", Information Sciences, 2005

ACL-MLNER 2003, Workshop on Multilingual and Mixed-language Named Entity Recognition, ACL 2003, Sapporo, Japan, http://acl.ldc.upenn.edu/acl2003/mlner

MUC-6 (1995) Proceedings of the Sixth Message Understanding Conference (DARPA), Morgan Kaufmann Publishers, San Francisco.

Paxson V. (1995) Flex – Fast Lexical Analyzer Generator. Lawrence Berkeley Laboratory, Berkeley, CA. Available at ftp://ftp.ee.lbl.gov/flex-2.5.4.tar.gz

Poibeau Thierry (2003) The Multilingual Named Entity Recognition Framework. In: EACL 2003 proceedings. pp. 155-158

Pouliquen Bruno, Steinberger Ralf, Ignat Camelia & de Groeve Tom (2004a). Geographical Information Recognition and Visualisation in Texts Written in Various Languages. Proceedings of ACM-SAC, pp. 1051-1058. Nicosia, Cyprus, 2004.

Pouliquen Bruno, Steinberger Ralf, Ignat Camelia, Käsper Emilia & Temnikova Irina (2004b). Multilingual and Cross-lingual News Topic Tracking. In: CoLing 2004 proceedings, Vol. II, pp. 959-965. Geneva, August 2004.

Shinyama Yusuke, Sekine Satoshi (2004), Named Entity Discovery Using Comparable News Articles, in CoLing 2004 proceedings, Vol. II, pp. 848-853. Geneva, August 2004.

Steinberger Ralf, Pouliquen Bruno & Ignat Camelia (2005). *Navigating multilingual news collections using automatically extracted information*. Proceedings of ITI'2005. Cavtat (Croatia), June 2005.

Swofford Mark (2005). http://www.pinyin.info/ and http://www.romanization.com/ (accessed on 26/05/2005).

Zobel Justin and Dart Philip W. (1995). Finding approximate matches in large lexicons. Software-Practice and Experience, Vol. 25(3), pp. 331–345

NOTES

1. Demonstration available at http://press.jrc.it/NewsExplorer
2. A search on Google gives an idea of the usage of each spelling as:
Mohammed: 7,410,000
Mohamed: 5,340,000
Muhammed: 848,000
Muhamed: 119,000
3. http://en.wikipedia.org/
4. The article did in fact intend to talk about Daniella Cicarelli ('reciente separación de la modelo brasilena Daniel Cicarelli'); last accessed on 1/06/2005 at http://www.lanacion.com.ar/deportiva/nota.asp?nota_id=708643&origen=rss.
5. $F = 2\frac{P \cdot R}{P + R}$
6. http://www.bbc.co.uk/worldservice/
7. http://www.graphviz.org/

ABSTRACTS

We present an exploratory tool that extracts person names from multilingual news collections, matches name variants referring to the same person, and infers relationships between people based on the co-occurrence of their names in related news. A novel feature is the matching of name variants across languages and writing systems, including names written with the Greek, Cyrillic and Arabic writing system. Due to our highly multilingual setting, we use an internal standard representation for name representation and matching, instead of adopting the traditional bilingual approach to transliteration. This work is part of a news analysis system that

clusters an average of 25,000 news articles per day to detect related news within the same and across different languages.

Nous présentons ici un outil de repérage des noms de personnes, à partir d'articles de la presse internationale, capable de reconnaître les différentes variantes d'un même nom. L'originalité de notre approche vient de l'identification des variantes de noms à travers les langues et systèmes d'écriture, grec, cyrillique et arabe compris. Étant donné notre contexte multilingue, nous utilisons une représentation interne standard de chaque nom ainsi qu'une même mesure de similarité (au lieu d'adopter l'approche bilingue habituelle de la translittération). Ce module fait partie d'un outil plus général qui analyse en moyenne 15.000 articles de journaux chaque jour, afin de regrouper les documents similaires, aussi bien dans une même langue que dans des langues différentes.

INDEX

Mots-clés: entités nommées, repérage multilingue d'entités nommées, translittération, traitement automatique (du langage), extraction d'information, multilinguisme

AUTHORS

BRUNO POULIQUEN

European Commission, Joint Research Centre, 21020 Ispra (VA), Italy - http://www.jrc.it/langtech – Name.Surname@jrc.it

RALF STEINBERGER

European Commission, Joint Research Centre, 21020 Ispra (VA), Italy - http://www.jrc.it/langtech – Name.Surname@jrc.it

CAMELIA IGNAT

European Commission, Joint Research Centre, 21020 Ispra (VA), Italy - http://www.jrc.it/langtech – Name.Surname@jrc.it

IRINA TEMNIKOVA

European Commission, Joint Research Centre, 21020 Ispra (VA), Italy - http://www.jrc.it/langtech – Name.Surname@jrc.it

ANNA WIDIGER

European Commission, Joint Research Centre, 21020 Ispra (VA), Italy - http://www.jrc.it/langtech – Name.Surname@jrc.it