Deliverable D15.5
Cross-linguistic IE tools for metadata discovery

DOCUMENT IDENTIFIER	PS_WP15_UTV_D15.5_CrossLing_IETools_v2.4
DATE	10/12/2007
ABSTRACT	This deliverable concerns the analysis of tools and techniques for cross-linguistic information extraction and retrieval as they are discussed in deliverable 16.3. Their aim is to support metadata extraction abilities for heterogeneous material, written in more than one language and in more than one style. The work provides a survey of the benefits of extraction tools for metadata discovery for the MAD subsystem.
KEYWORDS	Information Retrieval, Cross-language Retrieval, Information Extraction Text classification, Ontology-driven retrieval and browsing.
WORKPACKAGE / TASK	WP15
AUTHOR, COMPANY	Roberto Basili, Marco Cammisa, Alessandro Moschitti, Daniele Pighin Alfredo Serafini (RTV)
NATURE	Technical report
DISSEMINATION	PU : Public
INTERNAL REVIEWERS	G. Dimino (RAI)

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
0.1	2004-05-12	First Draft	Living	Confidential
0.2	2005-10-12	Second Draft	Living	Confidential
1.0	2006-01-09	First Version	Living	Confidential
1.1	2006-01-10	Second Version	Living	Confidential
1.2	2006-11-15	Third Version	Living	Confidential
1.3	2006-11-20	Fourth version	Living	Confidential
1.4	2006-12-01	First Full version	Living	Confidential
1.6	2006-12-15	Second revision of the Full version	Living	Confidential
1.7	2006-12-18	Third revision of the Full version	Living	Confidential
2.1	2007-02-18	Second Full version	Living	Confidential
2.3	2007-12-10	Final version	Final	Confidential
2.4	2008-04-09	Dissemination level changed to Public	Final	Public

1 Executive Summary

This deliverable concerns the analysis of tools and techniques for cross-linguistic information extraction and retrieval as they are discussed in deliverable 16.3. Their aim is to support metadata extraction abilities for heterogeneous material, written in more than one language and in more than one style.

The work provides a survey of the benefits of extraction tools for metadata discovery for the MAD subsystem. This will also guide the experimentation of IE tools over the results of content analysis (especially speech to text transcription) (as discussed in other deliverables (D15.6)). It will thus describe in detail the state-of-art of methods, technologies and software tools that implement/support cross-lingual forms of retrieval and extraction from textual material. Hereafter this document presents titles and expected contents of the sections as they would be found in the final version of the deliverable.

2 Motivations and Scope

Access to distributed information is a complex task for the heterogeneity of the sources and for the diversity of interests, expectations and purposes of the target retrieval processes. PrestoSpace is a typical scenario in this sense as

- **Heterogeneity** characterizes:
  - *Data typologies*, as source information characterize different media and different types of content
  - *Data formats*, as even the same media can be derived with different granularity and quality so that formats may highly vary across the archives
  - *Contents*, as the source information is not characterized by a single knowledge domain but is spread along heterogeneous semantic dimensions
  - *Languages*, as different structured archives may include data expressed in different natural languages
- **Expected uses** are quite heterogeneous as maintenance and dissemination of multimedia data are the extremes of a variety of possible application scenarios
- **The production of the source information and the access to such information are quite independent processes,** so that different retrieval methods should be applied when metadata of different nature are involved. Moreover PrestoSpace (and in particular the MAD activities foreseen for the overall system) is targeted to different communities of users (from archivists to final users) and different groups of content providers (from TV companies to museums) that are addressed with all or some of the offered functionalities.

In the domain of audiovisual archives of large TV broadcaster, four basic retrieval patterns have been identified:

- *Retrieving audiovisual items by information*. Starting from the specification of metadata constraints the material for which the stated constraints are valid is to be retrieved. This is the traditional use of the information as "metadata".
- *Retrieving information by audiovisual item*. The access to the archive information relies on the audiovisual material as the carrier of the pieces of information the users are interested in.
- *Retrieving information by information*. In this scenario, information is reached through the use of other pieces of information that act as "metadata" with respect to the target information.
- *Retrieving audiovisual item by audiovisual item*. Audiovisual material is sought and retrieved by means of similarity searches based exclusively on the audiovisual content, i.e. regardless of the expressed meaning and content.

The MAD activities of the past two years have been focused on a thorough analysis of these aspects. The conclusions suggest that the required information for the typical audiovisual archive exploitation process can be divided in the following fundamental classes:

- *Identification information*, e.g. titles, credits, program publication information.
- *Editorial parts information,* i.e. information about the relevant editorial sub-items of a program (e.g. news items).
- *Content-related information,* e.g. text of speech transcript, topics, descriptions, aural and visual low level descriptive features.
- *Enrichment information,* i.e. information coming from external sources generically or topically related to the program content

In the rest of this document we will concentrate on the methods studied and implemented in MAD about the two patterns of access referred as *Retrieving information by audiovisual item* and *Retrieving information by*

information as they represent the most advanced forms of access do not covered by traditional technologies (i.e. databases and traditional IR).

Differences among the multimedia materials involved, storage and retrieval habits and information access that characterize users groups represent the real challenge for the MAD retrieval technology. The issue of multilingualism here is also to be specifically addressed as most sources targeted by PrestoSpace are multilingual and cross-linguistic access must also be supported for the European dimension of the intended user community.

This document is organized as follows. Section 3 will synthetically describe the Information Extraction technology and its main cross-linguistic aspects. Section 4 will synthetically reports the major Information Extraction functionalities studied in MAD and it will use the overall architecture of a Semantic Analysis GAMP (the Italian one) as a reference. Section 5 describes a method for cross-linguistic information retrieval via query translation. Evaluation of the applied technologies is detailed in Section 6. A general discussion on the applicability of the integrated technologies to the Prestospace case is reported in Section 7, where some potential extensions are also discussed.

3 Information Extraction: Objectives, Methods and Technologies

3.1 Motivations

The MAD platform aims to exploit human language technologies for Information Extraction (IE) from the AV data made available by large archives. The nature and complexity of management, search and reuse of archive materials require complex storage and retrieval functionalities. These activities asks for:

- Recognition and indexing of *suitable generalizations* of relevant archive concepts as people names, organizations and locations
- effective retrieval functions that improve indexing at the simple textual level and support *conceptual rather than string retrieval*
- Interoperability at the levels of abstractions required by the AV contents. For example, AV data should be published, queried and exchanged in a distributed fashion. The development of Web publication should support distributed querying and semantic service-based instantiation and invocation. The semantic data descriptions are critical in these activities and interoperable models (ontologies) are needed.

Semantic Analysis is applied in MAD to fit such high-quality requirements from the available multimedia properties (e.g. audio) to suitable generalizations and ontological representation. In the Semantic Web area, these processes that goes from raw and textual data to ontological annotations are typically called Information Extraction.

The starting point for semantic analysis is thus the Automatic Speech to text transcription (ASR) of the AV data contents**.** Extracting text from spoken content of audiovisual material is a fundamental step allowing for several documentation tasks, as well as representing an important core of searchable data in the publication system. In the current set-up of the documentation platform an automatic speech-to-text engine is used, developed by ITC-IRST ("Istituto per la Ricerca Superiore di Trento"), capable of extracting text from English and Italian.

The redundancy that AV objects guarantee at the data level needs to be explored in order to govern the retrieval complexity at the proper quality. The problems due to noisy nature of the extracted data (e.g. errors in the ASR that produce mistakes in the grammatical recognition) should be properly limited. The aim is to make available to the overall extraction and retrieval components of MAD as much information as possible. In this perspective larger data sets should be taken into account than just the source AV data. The textual material in input should be processed and enriched of the following relevant evidences as semantic metadata:

- *Terminological* and *lexical information* local to the AV input data (via ASR)
- Recognition of citations to *Named Entities* (e.g. people or organization) from local data as well as from reachable external sources
- *Automatic computation of useful hyperlinks* between the archived AV data (e.g. the individual segments in broadcasted TV journals) and the distributed sources (e.g. Web-based newspaper portals and pages). These sources includes assessed textual descriptions of topics related to the AV segment contents and are trusted
- *Ontological information* contained in all the above sources, as representation of classes (e.g. geographical locations, organizations or persons), individuals (e.g. George Bush, or USA/United States) and topical classes (e.g. Education vs. Sport, Foreign Politics vs. Economics).

The extraction of this rich variety of information, required by MAD, is the target of specialized GAMPs called *Semantic Analysis GAMPs*. GAMPS in this class are language specific so that two SA_GAMPs have been designed for Italian and English information extraction respectively.

3.2 An Introduction to Information Extraction

Information extraction (**IE**) systems emerged in the late 1980's and early 1990's [Pazienza:1997] though forerunners date back to the late 1960's [Gaizauskas and Wilks:1998]. In contrast to IR, IE systems do not return the subset of documents from the collection deemed to be relevant to a given query. They are usually given a "template" definition of one or more specific concepts and a document collection, and return a set of filled templates. Templates, in the context of IE, are structured data representations, designed to capture attributes of objects and events predictably present in stereotypical occurrences in texts.

For example, in a *plane crash* we would typically expect to find the type of plane, the airline, number of passengers, flight origin and destination, and location and time of a crash. In this case, a template is usually designed to capture this information. The IE system would, given the template and a document collection, seek to fill an instance of the template for each plane crash event it detected.

Information Extraction is thus the process by which an automatic system is able to process documents in a linguistically motivated way and derive a structured representation of (part of) their content. It is to be seen as a process through which a *structure* is derived from unstructured and noisy texts. The IE technology has the key advantage over IR that the search for an event of a type for which a template has been defined can be more easily satisfied.

The IE process is usually applied in a batch fashion (*off-line*) to the document collection and it builds a large corresponding set of filled template as a structured database. The extracted database can be used when the user requirement is to find a similar event. The higher the level of abstraction provided by the template, the easier and more natural will be the interaction of the user with the updated template database. Notice how a template instantiated with details of the new event can be used to derive a series of database queries. They are effectively the same template but with one or more template slots replaced by variables to be instantiated in the search.

Widely explored fields of application range from the recognition and management of terrorist events ([MUC-3:1991], [MUC-4:1992]), of joint venture news ([MUC-5:1993]) to machine translation of meteorological bulletins, where translation of templates is applied rather than the more complex text translation.

One of the major subtasks of any IE process is *Named-Entity recognition*, as most of the traditional IE target information make explicit reference to person, locations that are seen as the participants to the event described by templates. For this reason automatic Named-Entity recognition (**NERC**) is by its own a typical IE task for which a large set of technologies has been defined and applied. Symbolic pattern recognition (e.g. [Appelt et al.:1993] and statistical methods [Bikel et al.:1999] are largely employed for NERC. The research over models of NERC and relation extraction has been recently conveyed on benchmarking and comparative evaluation by the NIST ACE challenge [ACE:2001]. The objective of the ACE program is to develop automatic content extraction technology to support the automatic processing of source language data. Possible down-stream processing includes classification, filtering, and selection based on the content of the source data, i.e., based on the meaning conveyed by the language. In the ACE 2005 [ACE:2005] challenge, five primary recognition tasks have been defined: entities, values, temporal expressions, relations and events.

There are two major limitations in general with IE technology. First, high performance, systems need to be hand-tailored for each new template/domain and this can be expensive. Secondly, template definitions limit the number of "questions" that can be asked. This means that the analyst cannot be expected to ask "ad hoc" questions that a particular topic might involve.

Notice that the IE approach is quite distinct from traditional ad hoc IR. The usual adopted document representation in IR is the simple *bag-of-words*. This is simply not applicable to IE where a more sophisticated pattern matching and reasoning ability is required and explicit forms of linguistic representation (e.g. subject-verb relationships) are needed. Moreover, especially in the Prestospace context, several problems arise in term of language translation when *bag-of-words* are adopted since they does not allow the system to rely on precise cross-linguistic information.

Given such problem, IR studies have been directed to the designing of more effective document representations that have relations with the IE tasks. Documents are still described as pairs *<feature,*

weight>, but, a more complex and effective feature design is applied. Such studies aim to achieve a representation more *conceptual* than the one provided by simple words. The consequences for multi-language applications are straightforward: a representation based on concepts rather than on words, limits all problems relates to the ambiguity. This allows the cross-language system to achieve the same retrieval performance of the monolingual systems.

Unfortunately, none of the advanced linguistic representations proposed in the literature for ad hoc IR tasks have been shown to improve optimal pure statistical approaches based on the simple *bag-of-words.* The major reasons for such failure are the following: (a) complex representations capture just a small piece of information more than the *bag-of-words* and (b) such representations are derived automatically, thus the errors introduced in the retrieval process compensate the poor gain in accuracy provided by the richer feature space.

Documents embody the highest expressions of the human linguistic skills. This intuitive observation has led researchers in document retrieval and information retrieval to consider linguistic aspects in the modelling of more complex document representations. Some of the well-known feature models experimented in the last decades are:

- ***Lemmas***, i.e., the base form of rich morphological categories, like nouns or verbs. In this representation, lemmas replace the words in the target texts, e.g., *acquisition* and *acquired* both transform in *acquire*. This should increase the probability to match the target concept, e.g., *the act of acquiring* against texts that express it in different forms, e.g., *acquisition* and *acquired*. Lemmatisation improves the traditional stemming techniques used in IR. In fact, the stems are evaluated by making a rough approximation of the real root of a word. As a consequence, many words with different meanings have common stems, e.g., *fabricate* and *fabric*, and many stems are not words, e.g., *harness* becomes *har*.
- ***Simple n-grams***, i.e., sequences of words selected by applying statistical techniques. Given a document corpus all consecutive *n*-sequences of (non-function) words are generated, i.e. the *n*-grams[1]. Then statistical selectors based on *n*-gram frequencies are applied to select those most *relevant* for the target domain. Typical used selectors are *mutual information,* χ^2 or *document frequency*.
- ***Nouns Phrases***, e.g., Proper Nouns and Complex Nominals. Simple regular expressions, e.g. N^+ (i.e., every sequence of one or more nouns), based on word categories (e.g., nouns, verbs and adjectives) can be used to select complex terms like *Minister of Finance* and discard the non-feasible term *Minister formally*. The words *Ministers* and *Finance*, in the first phrase, are often referred to as *head* and *modifier,* respectively. More modifiers can appear in a complex nominal, e.g., the phrase *Satellite Cable Television System* is composed of the tree nouns *Satellite*, *Cable* and *Television* that modify the head *System*.
- ***<head, modifier$_1$,…, modifier$_n$>*** *tuples*. Parsers, e.g., [Charniak:2000], [Collins:1997], [Basili et al.:1998] are used to detect complex syntactic relations like *subject-verb-object.* These can be used to select complex phrases, e.g., *Minister announces plans*, from texts. An interesting property is that these tuples can contain non adjacent words, i.e. tuple components can be words linked by a long distance dependency, e.g. in [strzalkowski-jones:1996] the *subject-verb* and *verb-object* pairs (i.e. the <*head, modifier*> pairs) were used. Hardly, such tuples can be detected via pure statistical models. The aim of phrases is to improve the precision on concept matching. For example, documents in an *Economic* category could contain the phrase *company acquisition* whereas an *Education* category could include term like *language acquisition*. If the word *acquisition* alone is used as an individual feature, it will not be useful to distinguish between the two above categories. The whole phrases, instead, give a precise indication of the document content.
- ***Semantic concepts***, each word is substituted with a representation of its meaning. Assigning the meaning of a content word depends on the definition of word senses in semantic dictionaries. There are two ways of defining the meaning of a word. First, the meaning may be explained, like in a dictionary entry. Second, the meaning may be given through other words that share the same sense, like in a thesaurus. For example, WordNet [Miller:1990] encodes both forms of meaning definitions. Words that share the same sense are said to be *synonyms* and in WordNet, a set of synonym words is called *synset*. The advantage of using word senses rather than words is a more precise concept matching. For example, the verb *to raise* could refer to: (a) *agricultural texts*, when the sense is *to cultivate by growing* or (b) *economic activities* when the sense is *to raise costs*.

[1] Traditionally, in IR *n*-grams refer to sequence of characters but they are also used to indicate sequence of words.

A typical IE engine employs a variety of the above information and require a complex workflow, where a cascade of modules support the recognition of part of complete portions of the previously describe features:

- ***Tokenization***, it mainly consists of recognition of token (word) boundaries and sentence boundaries;
- ***POS tagging***, it concerns part-of-speech (**POS**) tagging including lemmatisation, the POS Tagger assigns an appropriate part-of-speech as well as the appropriate lemma for each token (*cats* Nn-Pl vs. *cat* Nn-Sg); thereby multiword terms are also recognized;
- ***NE recognition***, it seeks to locate and classify terms and multiword terms into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. (e.g. "*Isaac Newton*" -> "*Person*", "*New York*" -> "*Location*");
- ***Chunking***, it analyses a sentence to identify the constituents (noun phrase, prepositional phrase, etc.), but does not specify their internal structure, nor their role in the main sentence;
- ***Parsing***, is analyses an input sequence in order to determine its grammatical structure, detecting the grammatical role of phrases (subject, direct object, etc.);
- ***Ontology Matching***, it finds correspondences between semantically related entities of an ontology. Every entity can be represented by a unique identifier and can have or inherit property (e.g. "*George Washington*" has been a "Person" and "*President in U.S.A.*").

Text categorization is the task to assign to a document, one or more categories/classes, based on its contents. Document/Text classification tasks can be divided into two sorts: supervised and unsupervised. The most common text classification applied that can be found in literature are:

- Naive Bayes [Tzeras and Artman:1993] is a probabilistic classifier which uses joint probabilities of words and categories to estimate the conditional probabilities of categories given a document. The naive approach refers to the assumption of word independence. Such assumption makes the computation of naive Bayes classifier far more efficient than the exponential complexity of a pure Bayes approach
- SVMs (Support Vector Machines) are based on the structural risk minimization principle: a quadratic programming technique finds the hyperplane in the space which minimizes the distance between the nearest documents of different classes and the hyper plane itself. This classifier has been successfully applied on academic benchmarks. It provides the highest performance on the Reuters corpus (about 86%). The problems arise when it is applied to operational scenarios where the number of training documents is thousands of times higher than the number of documents contained in the benchmarks. The disadvantage of SVMs is the training time, which can be very high if there are large numbers of training examples. Moreover, the classification phase can be very slow for non-linear SVMs [Drucker et al.:1999]. As the number of documents increases, the number of support vectors increases in a not well understood proportional law. This means that thousands of support vectors for assigning each single document could be involved. As each support vector requires a scalar product with the input documents, the time for an on-line classification is usually very high.
- CLASSI is a system that uses a neural network-based approach to text categorization [Ng et al.:1997]. The basic units of the network are the perceptrons. Given the amount of data involved in typical operational scenarios the size of the target network makes generally the training and classification complexity prohibitive. KNN is an example-based classifier [Yang and Chute:1994] which makes use of document to document similarity estimation. It selects a class for a document through a k-NearNeighbour heuristic. For this the algorithm requires the calculation of all the scalar products between an incoming document and those available in the training set. The optimisation proposed by the EXP-NET algorithm [Yang and Chute:1994] reduces the computational complexity to O(N ·log(N) time, where N is the maximum among the number of training documents, the number of categories and the number of features. The KNN time complexity is thus rather high
- Rocchio [Ittner et al.:1995], [Cohen and Singer:1999] often refers to TC systems based on the Rocchio's formula for profile estimation. An extension of the algorithm was proposed in [Schapire et al.:1998], [Lam and Ho:1998] but both approaches relevantly increase the complexity of the basic model. In PRC [Moschitti:2003] a parameterised version of the Rocchio classifier has been presented.

- RIPPER [Cohen and Singer:1999] uses an extended profile notion based on co-occurrences and multiwords. A machine learning algorithm allows the contexts(e.g. a windows of n words) of a word w to decide how (or whether) the presence/absence of w contribute actually to the target document classification. As it is based on profiles, it can be very fast in online classification task, but it has a noticeable learning time. Moreover, given the complexity to derive the suitable multiwords, it is not clear if it can be applied to millions of documents.
- Dtree [Quinlan:1986] is a system based on a well-known machine learning method (i.e. decision trees) applied to training data for the automatic derivation of a classification tree. The Dtree model selects the relevant words (i.e. features) via an information gain criterion and predicts the target document's categories according to word combinations. It efficiently supports online classification as the category assignment time is proportional to the time required to visit the decision tree.

The previous works describe the Text Classification task, applied on well written text. A special observation has to be carried out in the PrestoSpace framework where the text to analyse is not so well written , because it is automatically derived from an ASR (Automatic Speech Recognition) process. In the previous works, linguistic and syntactic information are used to obtain better performance. In the texts derived from and ASR, the syntax is not reliable, so with some changes the previous work can be applied without these information applying a simple bag of word model, obtaining reasonable performance.

3.3 Cross-Linguistic Aspects of IE

As already pointed out in Deliverable 16.3, the MAD publication subsystem should make use of human language technologies for Information Extraction (**IE**) from automatic transcribed speech and for robust shallow grammatical analysis of incoming AV data and for retrieval in any of the targeted languages.

It is to be stressed that redundancy at the data level is to be explored. The noisy nature of the extracted data (e.g. errors in the ASR or mistakes in the grammatical recognition) should be controlled by making available to the overall extraction system of source material as much as possible. In this perspective larger data sets should be taken into account than just the source AV data. The overall picture is that textual material in input should be processed by adding the following evidences as principled metadata:

- Terminological and lexical evidence local to the AV input data (via ASR when possible)
- Named-Entity recognition from local data as well as from external sources
- Automatically suggested hyperlinks between the target AV data to be archived (e.g. individual news in broadcasted TV journals) and distributed (and trusted) sources (e.g. Web-based newspaper portals and pages)
- Ontological information in terms of IR representation comprising ontology indexes about topical classes (e.g. Education vs. Sport, Foreign Politics vs. Economics), upper-level concepts (e.g. geographical locations, organizations or individuals) and specific instances (e.g. George Bush, or USA vs. United States)

This rich variety of information expected from semantic extraction in MAD poses then several requirements to Cross Linguistic Information Retrieval. First, the modelling of the user interface according to different (and integrated) querying capabilities:

- Full text search as usually applied by mostly popular search engines
- Natural Languages Questions
- Semantic access through the navigation of ontological information (i.e. concepts, relations and instances) and reference to the ontological indexes in the source AV documents

All the above information is to be intended in language neutral framework: full texts should be searchable in different languages, while ontological information as well as NEs should be properly represented so that language ambiguities and variability are taken into account.

Second all the above search modalities should be offered in a language independent fashion. The discussion of technological solutions to support the above processes is reported below in this document, as they have a relevant impact on the accuracy reachable by the PrestoSpace solutions to CLIR issues.

The viable solutions to the above problem concern

- The adoption of language neutral representation (via the KIM ontology)
- Query processing (expansion and translation) for dealing with multilingual information during retrieval

3.3.1 The KIM approach

A part of the cross-linguistic facilities available in PrestoSpace are carried out by means of the KIM system. The role that KIM plays for the enhancement of the functionalities of the MAD platform, relates to the IR process, and in particular for the purpose of retrieving digitised audiovisual material.

To perform IR, KIM makes use of basic upper-level ontology (PROTON) and massive KB. PROTON contains about 300 classes and 100 properties, thus covering wide range of the general concepts required for the semantic annotation, indexing, and retrieval of documents. PROTON has been designed with the intent to serve as a basis for the creation of various domain-specific ontologies. The KB, contains almost 80000 entities and their aliases (including aliases in several European languages). The entities have been collected from various sources like geographical and business intelligence gazetteers. The KB covers real-world entities that could be referred in content across a wide variety of domains (e.g. Locations are important for News, Tourism, Documentaries, etc). The KIM World KB contains entity descriptions that represent a named entities in terms of their aliases, relations to other entities, attributes and the entity's proper class.

Semantic IR needs a specific form of information extraction based pre-processing called semantic annotation. KIM analyses texts and recognizes references to entities (such as persons, organizations, locations, dates). Then it tries to match the reference with a known entity that has a unique URI and description. Alternatively, a new URI and description are generated automatically. Finally, the reference in the document is annotated with the URI of the entity. The metadata, created in this way can be used for indexing, retrieval, visualization and automatic hyper-linking of documents. Therefore, semantic annotation, as performed by KIM, is the generation of specific metadata. It is the process of assigning to the entities in the text links to their semantic descriptions in the KB. It is important to mention that the semantic annotation process is applicable to any sort of content that could be used with traditional information extraction techniques: web pages, documents, text fields in databases, transcripts of news emissions, etc.

One of the specifics of semantic annotation is that it may provide more precise information as to the NEs type than the systems based on flat NE type sets lacking taxonomy or other sort of definitions would. While the result of the traditional NE recognition approach gives only few basic types the entities might belong to, for example:

<Person>Lama Ole Nydahl</Person>,

After a semantic annotation has been performed, more specific NEs type is also given:

<ReligiousPersonID="http://..kim/Person111111">Lama Ole Nydahl</ReligiousPerson>

Having semantic annotation over the content, KIM can index the content with respect to the mentioned entities. The metadata is used as an index pointer for the respective entity during the retrieval process. Because of the specifics of indexing that KIM provides, new (semantically-enhanced) access methods (or user-need definitions) are enabled. The user could specify queries consisting of constraints about the types of the entities, relations obtaining among entities, and/or entity's attributes. This means that the user could specify the NEs to be referred to in the content of interest, using name restrictions (e.g. a Person whose name ends with '*Alabama'*). An example of a query consisting of pattern restrictions over entities is:

> Give me all materials referring a Person that hasPosition "*CEO*" within a Company, locatedIn a Location with name "*UK*".

To answer the query, KIM applies the semantic restrictions over the entities in the instance base. The resulting set of entities is matched against the index, produced by the semantic indexing of the processed materials. Then the referring content is being retrieved with relevance ranking according to these NEs. Queries of this kind could also be combined with conventional keyword search (full text search, **FTS**) and thus, could benefit from the combination of both approaches (e.g. via intersection or union).

To illustrate the usefulness of semantic IR as compared to classical IR, consider the following example: the user attempts to retrieve the documents containing information about a telecom company positioned in

Europe. The IR of usual sort cannot retrieve the materials about *Vodafone*, because it cannot link *Vodafone* to the description "*a telecom in Europe*", namely because it cannot infer that *Vodafone*, being a mobile operator, is a kind of a telecom; or that *Vodafone* being positioned in UK, means it is positioned in Europe, etc. Having the background information in the ontology and the instance base, and semantically-enabled information extraction modules, the semantic retrieval techniques would produce results that are superior to the ones of traditional IR in two ways. Having precise proper class information (e.g. that an entity is `CommercialOrganization`) searches like "*give me materials about Commercial Organizations that have ImpEx in their name*" would result only in materials that contain such organizations, while a FTS with the keyword *ImpEx* would result in all materials that contain the phrase regardless of the entity type associated (if any). Another scenario is when you search for an entity but you specify only one of its names, like *Beijing* and not *Pekin*. A FTS for *Beijing* would return just documents containing exactly this alias of the city, while the semantic IR would consult the instance base and implicitly expand the query to all the aliases of the city. This is not done explicitly since the name itself is not used as a pointer to the indexed content, instead KIM uses the entity ID which is the same no matter which alias has been used *Pekin* or *Beijing*.

3.3.2 Cross Language Information Retrieval as query expansion and translation

3.3.2.1 Query Translation as a Wordnet-based expansion process

In the scenario foreseen in the documentation and publication processes in PrestoSpace also textual (i.e. keyword based) or natural language queries must be supported. The query is given in one language (e.g. English), and the AV items to be searched may be expressed in a different language (e.g. Italian). A typical example can be the query "*Iraqi war"* that, in order to properly trigger the retrieval process, should be translated into "*Guerra in Iraq*" as reportages or news about the conflict (*conflitto* in Italian) are meant to be the target.

As discussed in previous sections, this CLIR task is usually tackled by statistical or knowledge based approaches. The first are based on large corpora of aligned texts, i.e. texts in both languages, obtained via direct sentence translation. They show good precision and recall performances but they are strictly tight to the availability of parallel texts. Moreover, they are highly sensitive to domain variations, making thus hard their portability among domains and collections (i.e. different and heterogeneous AV data streams). Dictionary or Knowledge based approaches have a lower accuracy, but do not require training data. They are currently based on bilingual dictionaries. However, they are even more sensitive to domain variations. The result is that they are also difficult to adapt to the dynamically changing application domain. For example, Google does not use this technology.

3.3.2.2 Semantic Disambiguation.

Recent work in word sense disambiguation suggest that any statistical approach to disambiguation should account for the significant differences that characterize sense distributions (i.e. probability distributions of senses given the words) across different domains. Predominant (i.e. most likely) senses in one domain (or collection) do not generalize, and change from one test scenario to another.

[McCarthy et al.:2004] proposes a method to estimate predominant senses via a large untagged corpus, reaching high accuracy on standard WSD benchmarks. The underlying assumption is that predominant senses are not stable across domains and corpora, and that the notion of predominance can be modelled through an unsupervised corpus based analysis. In that work also syntagmatic evidence is taken to build thesauri of semantically similar words, that in turn results in an effective modelling of the proper sense probability distributions. The main limitation of that approach is that the sense distribution is modelled once for all contexts on the basis of the corpus adopted for unsupervised learning, However, the notion of predominance is even more dynamic than the one suggested by McCarthy and colleagues. It is certainly given by the entire corpus as it embodies most of the agreement among writers and readers regarding the semantic contribution of individual lexical items. The corpus should thus enter in process by providing source evidence to estimate "global" sense probabilities. However, predominance also depends on individual occurrences of target words, e.g. document sentences or paragraphs, or the shorter text chunks expressing the queries.

A significant step further in disambiguation would be to capture both the global evidence embodied by the corpus as well as the implicit domain local to each context. Each sentence in fact suggests a topical context where the semantics of target word is highly constrained. When such topical context is made available the disambiguation process is easier as it can exploit both evidences.

While estimating most likely senses given the global or local domain is a problem that involve metrics of sense similarity and distances, the ways of capturing local domain evidence is a different task aiming to explore combination of several sources as a source of semantic evidences. The former aspect will be briefly discussed in the immediately following section. The ways of capturing local domains is the focus of the next section.

3.3.2.3 Unsupervised semantic similarity estimation.

A large area of research related to semantic disambiguation is represented by the deep work made on word sense disambiguation **WSD** (e.g. [Lesk:1986], [Schutze and Pedersen:1995], [Yarowsky:1992], [Yarowsky:1995], [Ng:1996], [Agirre and Rigau:1996], [Mihalcea:1999], [Dagan:2000], [Buitelaar:2001], [Pianta et el.:2002] , [McCarthy et al.:2004], [Gliozzo:2005]). Most work has been done on ways of estimating semantic similarity between word (pairs) according to distance metrics based on resources (e.g. dictionary definitions as in [Lesk:1986], or lexical hierarchies as in [Agirre and Rigau:1996]) or on corpora (e.g. [Yarowsky:1992], [Gliozzo:2005]).

A well known method is based on the notion of conceptual density [Agirre and Rigau:1996], [Basili et al.:2004]). Conceptual density (CD) measures the similarity among target words as a function of the informational utility of a lexical hierarchy able to represent (i.e. subsume) most senses of the targets. It depends on the topological structure of the underlying lexical semantic network and given the target words results in the selection of one or more sub-hierarchies generalizing most of the target word senses. It results at the same time in:

- (explanatory outcome) a number of generalizations (higher level senses) of lexical word senses useful to subsume (i.e. explain) the target words
- (quantitative outcome) A quantification of the quality of the different reachable generalizations according to the conceptual density metrics: the most dense if the hierarchy wrt to the source words the higher is the score. This score can be easily interpreted as a similarity score and give also rise to a distance metrics. A probabilistic interpretation of the scores allows to trigger sense disambiguation as a statistical task ([Basili et al.:2004]).

In [Basili et al.:2004] efficient algorithms for the derivation of both explanatory and quantitative information about the best generalizations as they can be found in Wordnet [Miller:1991] are presented. Notice how one of the advantages of this metrics is that it applies to pairs as well as *n*-ary sets of words. Moreover, CD only depends on the network structure and does not require any training over labelled examples.

The source information of this method is a set of associated words *w'* that should suggest the proper lexical sub-hierarchies for disambiguation, i.e. isolating the best senses and neglecting the odd senses irrelevant for the target set. Given a target noun *w* (e.g. a noun in a query) ways of building the target set for trigger the conceptual density disambiguation method may depend on

- *Syntagmatic evidence*, i.e. grammatical properties local to the context of *w*, e.g. the syntax of the query.
- *Paradigmatic evidence*, e.g. the topological properties of the ontological context for *w*. If *w* is the name of a concept *Cw* in an ontology, aliases *w'* of *Cw* or the names *w'* of super-ordinates or sub-ordinates concepts of *Cw* can be added to the target set. The resulting outcome of the CD method can be seen as an explanation (disambiguation) of *w* as the name of *Cw*.
- *Associative evidence*, as words *w'* can be collected from those contexts surrounding each occurrence of *w* in an underlying corpus.

The above properties of the CD method makes it appealing for query translation. For each word w of the query in fact, a specific target set can be derived with the above mentioned different methods. For example, given the noun *conflict*, and a query like "*reportages about Iraqi conflict",* syntagmatic or associative evidence can be collected outside the query[2]to expand this word into a target set like: {*war, battle, enemy,*

[2] In the next section we will show a model for acquiring such an expansion from unlabelled thematic corpora.

army, …, } expressing other synonyms, syntagmatically equivalent or topically associated words. When CD is applied to the derived target set it will converge more easily to the proper sense 3 of the conflict word, i.e. *"a hostile meeting of opposing military forces in the course of a war",* rather than the irrelevant sense "*opposition between two simultaneous but incompatible feelings*". Only in this case the proper translations (i.e. *conflitto* e *Guerra*) can be activated and added to the target language (i.e. Italian) query.

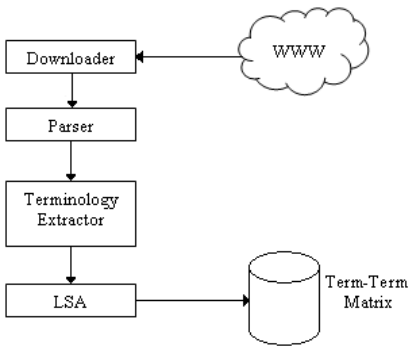
An important consequence of sense preferences obtained over WordNet in the above manner is that individual senses (i.e. synsets in the American English, i.e. Princeton, WordNet) can be mapped in other languages as well, given the availability of multilingual extensions of WordNet, called Multiwordnet [Pianta et al.:2002]. This is thus not only true for the English-Italian language pair (of interest in PrestoSpace) but also for a variety of other languages. Individual senses, i.e. synsets, are mapped across languages so that Italian names (i.e. synsets expressed into Italian nouns) are available: the selection of a synset in one language can easily trigger the extraction of corresponding nouns in the target language, as a form of translation via query expansion.

3.3.2.4 LSA-based domain modelling.

For each target noun *w* in a query, the unsupervised similarity metrics over WordNet, as suggested in the previous section, requires a target set, i.e. a set of nouns *w'* semantically associated with *w*. Associative evidence occurs when lexical entries (like conflict) are mapped into sets of topically related words, like *war, battle, enemy* or *army.* Notice that we are not looking necessarily for paradigmatic relations, like for example synonymy. The notion of semantic field here can be useful as the example demonstrate: the overall set *war, battle, enemy* or *army* constitutes a semantic field of *war*. Target sets should provide us with information for disambiguation and semantic fields can play here a role. In the example they enable to catch the proper translation more easily than in a more complex reasoning or statistical manner. If a method to map individual occurrences of words (e.g. queries) to semantic fields is available, than a lexical description of a semantic field would be accessible and CD disambiguation would make its work.

The notion of semantic field is rather close to the notion of local semantic domain discussed before. As Latent Semantic Indexing **LSI** methods suggest, a local domain can be obtained as regions in the **LSA** space derived by **SVD** (Singular Value Decomposition). The semantic effect of the LSA method is to generate a space where terms and documents are defined in the same space. The virtual document given by a query as well as the target nouns (to be translated) when mapped via LSA transformations, is a point in the LSA space and is thus close to other semantically related terms. The surroundings of such region can be assumed to be populated with semantically correlated words. It is thus a useful model for the notion of the target set needed for the CD estimation. LSA-like analysis can then be integrated with CD as a preparatory step to derive associative lexicons. First SVD decomposition of the source term-document matrix is run by deriving the linear transformations required for mappings. Then for each query, a virtual document in the transformed LSA space is derived. The sets of terms close enough to it are computed and added to the target set. CD disambiguation can then be run over the resulting lexical set and sense preferences are derived.

To create the matrixes term-document for Italian and English languages in the target domain of news provider, different sources has been chosen. For Italian, has been chosen the following news provider reachable from the web: "La Repubblica" and "RaiNet"; for English: "The Guardian" and "CNN". The pages collected has been parsed and the terminology extracted, so to obtain two term-document matrixes.



The LSA method appears promising in the PrestoSpace application, as the SVD decomposition, it is based on, can be applied without much effort:

- It does not require any annotated corpus as it runs over the simple term-document matrix
- It can be fed with information derived from any collection, and in particular with the AV document already available (i.e. ASR from video or audio programs).
- It is robust with respect to errors in transcriptions
- It is language neutral so that it can be applied to material in different languages

3.3.2.5 Dynamic Domain-driven Query Translation.

The result of the integration of the above methods is a fully unsupervised process of query translation whose applicability to PrestoSpace-like scenarios is very high. First, it consistently exploits the wide available resources of semantic evidence (i.e. corpora dealing with the user/application domains of interest), so realizing a powerful model of query translation *on-the-fly*. The scale of the AV archives targeted by MAD methodologies in PrestoSpace guarantee the availability of such large collections of (ASR transcribed) AV material. They are thus able to consistently trigger LSA modelling as well as to estimate sense disambiguation probabilities with precision.

Second, the methodology is highly portable throughout domains and applications as it combines learning methods that are fully unsupervised. This is thus ideal for a technology like PrestoSpace where heterogeneous archives are targeted. For this reason it represents an effective (and efficient) approach that should be explored in PrestoSpace.

3.4 Information Extraction vs. Information Retrieval

The rich variety of information extracted by the GAMPs in MAD poses then several requirements to the Information Retrieval functionalities in the publication phase. First, the user interface should model access methods according to different (and integrated) capabilities:

- *Full text search* as usually applied by mostly popular search engines
- *Natural Languages Questions*
- *Semantic browsing* as navigation through concepts, relations and instances of the ontology

All the above functionalities are to be intended as language neutral: full texts should be searchable in different languages, while ontological information as well as NEs should be properly represented so that language ambiguity and variability are taken into account. Second all the above search modalities should be offered in a language independent fashion. The discussion of technological solutions to support the above processes is reported below in this document, as they have a relevant impact on the accuracy reachable by the PrestoSpace solutions to CLIR issues.

The viable solutions to the above problem concern

- The adoption of *language neutral representation* (via the KIM ontology)
- *Query processing* (expansion and translation) for dealing with multilingual information during retrieval

Notice how the IE GAMPs in Prestospace allow to derive the following types of information as semantic metadata in the current EDOB structure:

- Terminological and lexical evidence local to the AV input data (via ASR when possible)
- Named-Entity recognition from local data as well as from external sources

- Automatically suggested hyperlinks between the target AV data to be archived (e.g. individual news in broadcasted TV journals) and distributed (and trusted) sources (e.g. Web-based newspaper portals and pages)
- Ontological information in terms of IR representation comprising ontology indexes about topical classes (e.g. Education vs. Sport, Foreign Politics vs. Economics), upper-level concepts (e.g. geographical locations, organizations or individuals) and specific instances (e.g. George Bush, or USA vs. United States)

It has been our design choice to capitalize all the possible information derived from the above metadata to support the search processes triggered by the user of the MAD Publication platform.

The information units searchable in the Publication platforms are entire programs, likely represented by means of an entire EDOB structure, as well as more fine grained units, like individual news segments in broadcasted news. These latter are resulting from the editorial object segmentation produced during the documentation chain. Once the elementary and comprehensive units are obtained all metadata associated with the individual units are available and they constitute informative indexes for the units. Redundancy can be here exploited as, for examples, some persons are mentioned more than once and the topical categories (e.g. sport vs. politics) are also known. Moreover, ontological Ids are associated as metadata and they constitute special indexes: first, they are language independent as the "name" of the index is the same for all languages; second, this metadata are also normalized across several mention types, e.g. *G.W. Bush* vs. *the president Bush*. Notice that patterns of co-occurrences can naturally emerge from the data as news associated with a topic are also including frequently some person names or some ontological Ids. This augment the ranking capabilities of traditional forms of IR, like vector space models adopting cosine similarity as the estimation of relevance between a unit and a query. This last point is crucial here. In Prestospace the variety of information available for individual units provides a rich representation where available evidence ranges from topics to words and ontological concepts. As a consequence, the representation of queries should be equivalently rich: the richer evidence can be extracted from a query the better will the quality of results from simple relevance ranking models. This is especially true when specific individual news (i.e. elementary units) are targeted.

The representation model adopted in Prestospace is driven by the KIM retrieval engine and it includes specific indexing at the level of individual metadata associated with an EDOB or with an EDOB segment. First of all entire programs are indexed separately from their individual units, and the search modalities will distinguish these two cases: when searching entire programs the user often use program identification properties (e.g. TV channel and dates), as opposed to specific kinds of search more focused on content issues, like person names or events and topics. Second, indexes are typed as different metadata give rise to different classes of indexes: simple textual features like common nouns or verbs will be represented separately from the Named Entities or from the ontological Ids. In this way all the different information are helpful even in cases when partial recognition has taken place: for example, missing recognition of ontological entities, (e.g. *Prodi* as a Named Entity that is not connected with the unique identifier of *Romano Prodi* in the ontology). Third an equivalent indexing is also applied to natural language queries. A topical categorization of short queries in fact can be usefully applied to distinguish the domain of interest of a given query, e.g. *Berlusconi speech in the Italian Parliament*, i.e. Politics, vs. *Il Milan chiede a Berlusconi nuovi giocatori*, i.e. Sport/Football. Also Named Entities and ontological entities are recognized according to the same technology adopted by the SA_GAMPs over the news texts. In this way the query is mapped into a (structured) vector of topics, terms, Named Entities and Ontology identifiers. It is worth noticing that in this scenario longer queries with are preferable: more details in fact can be introduced and they are helpful to improve the quality of the overall extraction applied to the NL query. Topical categorization in fact is more performing over longer queries rather then on shorter one. The More information is available the better will be the vector representation. This typed structure can be employed differently during retrieval. In a first way, types can be individually matched in the archive indexes, so that separate types provide independent matches: they contribute separately to the relevance estimation function. A simpler model would express the different types as different weights in the query vector: for example, Named Entities (that are likely to be more representative of a news item content) can receive higher weights in order to increase the relevance of the EDOB segments that include them. Weighting differently the types is a key criteria to adapt, in a simple manner, a traditional vector space model to the search requirements of the Prestospace rich metadata representation. Notice how different choices (i.e. different weighting policies for individual types) can be designed according to the different application scenarios: topical information can be more important for documentaries while Named Entities are better characterizing the contents of broadcasted news. These parameters represent a further flexibility of the overall IR approach in the MAD Publication platform. Notice how the IR indexes built according to the Semantic Analysis can be stored in a completely independent

fashion from the EDOB archive (EMS storage resource) so that retrieval can be as efficient as in large scale bibliographic DBs.

The treatment of multilingual issues and some examples of queries and query vectors are reported in Section 3.3.2, 5 and 6.3.

4 A typical platform for IE from multimedia data

4.1 Information Extraction in MAD

In the PrestoSpace framework, the MAD group should provide methodologies and methods for the extraction and indexing of metadata from multimedia sources. In particular a text stream, split in segments, can be derived from an automatic speech recognition phase. Segment are derived from media different from the simple text (e.g. video changes and silences in the audio channel).

These segments of text can be analysed using classical NLP techniques, even if this stream is not syntactically well formed. Due to this limitation special attention has to be paid when classical NLP methodologies are applied. As an example consider that several methodologies applied, for the Named Recognition, in literature use the capitalization of words and punctuation, that in this framework cannot be applied.

After processing the audio input, text-processing tools operate on the text stream produced by the ASR subsystem and perform the following tasks:

- Named Entity Recognition (NERC)
- Topic Classification
- Web Alignment
- Ontological annotation (via KIM)

The multimedia nature of the data, require that a complex data structure has to be used to represent and maintain the metadata extracted from the above linguistic components. In the *PrestoSpace* framework the "*Edob*" (Editorial Object), is used for the preservation of these metadata (linguistic driven derived from the text) and also not linguistic (i.e. content driven derived from the video stream, by the *"Content Analysis GAMPS"*).

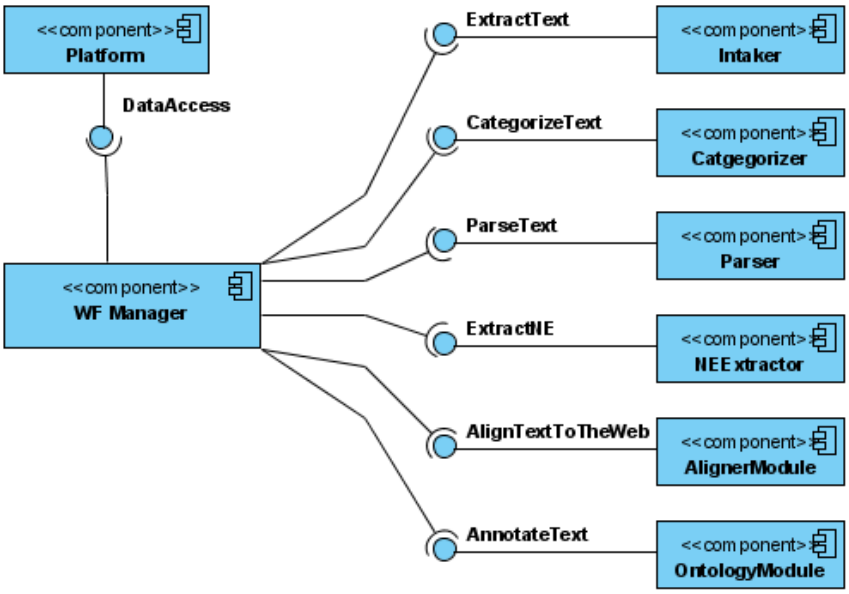
4.2 A general architecture for IE for multimedia data

As described in section 4.1, a cascade of processes are used to enriched the multimedia data with metadata. As an example, the Italian IE architecture will be described.

All the enrichment processes have to be organized (or synchronized) by a specific module called *"Workflow Manager"*. This module calls the processors according to their dependencies. Without loss of generality, the modules called are:

- Intaker
- Categorizer
- Parser
- Named Entity Recogniser
- Aligner
- Ontology module

The Information Extraction chain first applies the "*Intaker"* module. It collects and normalizes the incoming broadcasted news items as they are transcribed and segmented by the speech recognition tool. Then, the *Categorizer* is invoked by the Workflow Manager over the input news items: it returns the pertinent topical categories (with their associated confidence scores) according to a classification scheme that has to be defined previously. In the Italian semantic analysis Gamp the RAI internal classification scheme has been applied. Concurrently, these news items can be parsed (via the *"Parser"*) to detect Named Entities (via the *"Named Entity Recogniser"*), these provide a set of significant metadata. These can be used by the *Aligner* module to search candidate's news items that are similar to the pages downloaded by a *Web Spider*. The retrieved Web pages are also parsed[3] and indexed according to traditional IR techniques. For each news item, the *Alignment* process selects the suitable Web pages from the set of the retrieved candidates and sets direct hyperlinks to them. Theses links are used to include further (external) metadata, auxiliary to the internal ones to improve the overall accuracy due to wrong or irrelevant information[4]. Finally a module that uses an Ontology to annotate the news item is applied (in PrestoSpace, KIM as Knowledge & Information Management platform as been used). This provide a language independent representation for Named Entities. As an example consider that the *"White House"* (the government building that serves as the residence and office of the President of the United States), is usually translated in other language (e.g. in Italian the correct translation is *"Casa Bianca"*). The ontology represent this entity as a single item identified by an id (i.e. an Uniform Resource Identifier *"URI"*), that is for its nature language independent.



The above figure shows the overall architecture. It can be noticed haw the "WF Manager" plays the central role of coordinator between the other components. It decides the right sequence of calling and also manage all the unpredictable errors that can occurs. Can also be considered as a gateway between the external environment represented by the "Platform" and the components that provide the effective functionalities.

The architecture is in this way flexible in respect to add other functionalities, in fact other module or NLP processor can be easily added in the cascade of processes.

[3] The parsing process is different in the two cases as automatic transcriptions follow less rigidly linguistic well-formedness criteria so that specific grammatical and lexical rules are required.

[4] When mistakes made by the speech recogniser over incoming transcriptions affect the quality of the source metadata, external, i.e. Web originated, metadata can be used to validate the former and compensate such errors.

4.3 Evaluation Aspects

4.3.1 Definition of Performance indexes and Experimental Set-up

Objective measures in IR have been traditionally employed for the quantitative evaluation of accuracy and robustness of retrieval functionality. The traditionally adopted functions are *precision* and *coverage* (also called *recall*) and they are usually computed over controlled data sets. These data sets work as oracles for the target IR system and are specifically tailored to the target applications. These sets are built according to a reference collection and they include controlled queries together with their related document, i.e. the documents of the collections that are relevant to each individual query. A quantitative description of the relevance can also be given by the oracle: for example, some collections include a graded relevance score that organizes relevant documents into levels (from the most relevant to the relatively irrelevant documents).

Different results are usually obtained when these measures are applied, on a given IR system or model, to different data sets. The superiority of a methodology with respect to others has to be established by investigating how well the system performs on data derived from different collections: the closer (i.e. significantly overlapping with or totally immerse in) they are with the data over which the final system will work, the more precise will be the evaluation. The activity of producing controlled material, i.e. oracles for the different sub-tasks, is very costly but it is the only way to derive objective estimations of the realistic system performances in real scenarios.

The kind of capabilities expected for the CLIR system in MAD are very different. They range from AV data topical categorization to Web alignment; from retrieval of individual news items to automatic generation of hyperlinks, to query translation. Each of the above mentioned tasks will require a careful specific analysis of performances, coupled with error analysis for fine-tuning in the last phases of the project. For example, mistakes in the ontology-based retrieval may depend on lacks in coverage of the available Knowledge Bases (i.e. missing instances of some concept or missing properties or relations in the ontology itself), in problems of the decision-making algorithms required to deal with ambiguous cases or in misleading annotations produced over AV data sets by the IE algorithms. Similar problems may arise in cross-linguistic information retrieval where query translation may fail due to lacks in the Wordnet lexical coverage or in errors in the sense disambiguation algorithms (see section 5.2.1). As the error analysis is essential for the tuning of the MAD system before the end of the project lifecycle, the size and quality of the controlled data sets for the different activities is a critical issue for the final success of the project.

It is thus suitable that specific effort is spent by the technical groups involved in the CLIR activities as well as by the user groups involved in MAD to build extensive and reliable controlled (i.e. annotated) data for performance evaluation and error analysis. Specific collections should thus be built as repositories from which specific oracles can be extracted. These will involve: oracles for the benchmarking of the full text retrieval (i.e. sets of queries with the corresponding correct answers as individual AV items satisfying the requests); oracles for the ontology based retrieval as complex queries based on the available concepts and relations coupled with exact answers; oracles for cross-linguistic retrieval including sets of queries in the source language with AV answers in (possibly more than one) target language. Notice that specific parallel corpora should be created made by multimedia material in different languages insisting on the same domain and events: for example a collection of TV broadcasts from RAI and BBC over the same time period could be used as a starting point both for training and testing CLIR models. A specific attention should be also devoted to methods for evaluating automatic hyperlinking capabilities among individual AV items as well as Web documents, as expected in the MAD semantic analysis. Protocols for evaluating the quality and usefulness of individual links should be defined and, accordingly, test material should be created for quantitative evaluation.

Although the costs of the specific development of test material is high, the resulting resource would be invaluable in light of the technical assessment internal to the project as well as a general guideline and reference for the future development of the multimedia data indexing and delivery technology in Europe.

4.3.2 Measures of performances for IR

In Information Retrieval, several accuracy measures have been proposed as the task of evaluating system decisions is only vaguely defined and comparative evaluation among systems is difficult. The development of reference collections gave the possibility of defining objective quantitative measures, each one with inherent advantages and disadvantages. Usually these measure are applied to system decisions and are equivalently valid for tasks like ad hoc document retrieval as well as text categorization (Basili and Moschitti, 2005). As every system decision (e.g. Named Entity Recognition and Classification) is seen here as a measurement point also other tasks, not traditionally inspired by IR, are covered by the same measures. For this reason, we will describe all the measures within a document categorization task, where a single system decision is the assignment of a document to a specific class: the oracle (the reference collection also called golden standard or test set) reports if such a category is correct (acceptable) or not for the underlying documents. More than one class may be still valid for a single document and the different document class pairs <*d,c*> are considered as different and independent decisions in the following discussion.

The *error rate* is the ratio between the number of documents not correctly categorized and the total number of documents. According to the above definition, if the test set includes a small percentage of documents labelled under a given category, a trivial classifier which rejects all documents of that category will obtain a very low error rate (i.e. a good performance), at least with respect to that category. Two other measures, i.e. *Precision* and *Recall*, are not affected by such limitation. Given a specific category C_i, a decision function *h* that given a document *d* produces a set of classes suggested for *d*, an oracle *GS* that establishes the valid classes for each *d*, their technical definition can be stated in terms of three quantities:

- (*True Positives*) the number of correct documents, a_i, found by the decision function *h*, i.e. the number of documents in the test set *TS*, $d \in TS$, such that $C_i \in h(d)$ and $C_i \in GS(d)$.
- (*False Positives*) the number, b_i, of incorrect documents, i.e. the number of documents $d \in TS$ such that $C_i \in h(d)$ and $C_i \notin GS(d)$.
- (*False Negatives*) the number, c_i, of documents not retrieved for a class, i.e. the number of documents $d \in TS$ such that $C_i \notin h(d)$ and $C_i \in GS(d)$.

The *Precision* and *Recall* scores are defined by the above counts:

$$Precision_i = a_i /(a_i + b_i)$$

$$Recall_i = a_i /(a_i + c_i)$$

Both scores depend on inference policies, e.g. they depend on the threshold discussed in the previous section, and are in general inversely proportional.

When the acceptance threshold σ_i for a class increases, Precision also increases while Recall tends to decrease and vice versa. This variability between Recall and Precision makes difficult to compare two different classifiers: one could reach the highest Recall while the other could achieve the highest Precision. Thus, to get a single performance measure, the *Breakeven point* (*BEP*) is widely adopted. *BEP* is the point in which *Recall* and *Precision* are equal. It is estimated iteratively by increasing the threshold from the value of *Recall* = 1 (e.g. $\sigma_i = 0$) until $Precision \leq Recall$. The major problem is that the correct BEP score could not exist (i.e. for no value of the threshold *Recall=Precision*). In this case, a conclusive estimation is the mean between the *Recall* and *Precision* (interpolated BEP) such that $|Precision - Recall|$ is the minimum with respect to all the threshold values. However, even this may result artificial [Sebastiani, 2002] when *Precision* is not enough close to *Recall*.

The *F_1-measure* improves BEP definition by imposing the harmonic mean between *Precision* and *Recall* as follows:

$$F_1 = 2 \cdot Precision \cdot Recall / (Precision+Recall)$$

F_1 outputs a more reliable value especially when *Recall* is highly "different" from Precision. For example, with a *Precision* of .9 and a *Recall* of .001 the interpolated BEP is the average, i.e. 0.45 while the F_1 is 0.002. This latter corresponds to a more realistic performance indication. However, when comparable *Precision* and *Recall* values are obtained, the BEP drawback is not critical, thus it was used in [Yang, 1999; Joachims, 1998; Lewis and Gale, 1994; Apt´e et al., 1994; Lam and Ho, 1998].

Finally, classification problems usually involve more than two categories, thus we need a global measure to evaluate the performance of a category pool. In text categorization, the *microaverage* over all categories is

traditionally adopted. According to the definitions given for *Precision* and *Recall*, the following equations define the microaverage of *Recall* and the microaverage of *Precision* for a pool of *n* binary classifiers.

$$\mu Precision = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + b_i}$$

$$\mu Recall = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + c_i}$$

The above measures are then used to evaluate the *microaverage* of both *BEP* and F_1 , i.e.:

$$\mu BEP = \frac{\mu Precision + \mu Recall}{2}$$

$$\mu F_1 = \frac{2 \cdot \mu Precision \cdot \mu Recall}{\mu Precision + \mu Recall}$$

5 Cross-linguistic IR for MAD

The MAD (Metadata, Access & Delivery) Group, aim to develop a platform for annotate a retrieve metadata for multimedia and television broadcast archives. The mission of MAD system, within the wider objectives of the PrestoSpace factory, is to generate, validate and deliver to the archive users, metadata created by automatic/semiautomatic information extraction processors. These tools include audiovisual content analysers, automatic speech recogniser (ASR) and semantic analyser of the text extracted by the ASR. The MAD publication platform provides then access and search facilities to the exported metadata through an interface, which offers:

- Full Text Search on the metadata
- Ontology based browsing
- Content browsing
- Multilingual retrieval via natural language queries (NLQ)

The CLIR (Cross Linguistic Information Retrieval) engine, provides the functionalities needed to guarantee a multilingual retrieval of the metadata. The interaction with the user is provided by a natural language query mechanism.

The Publication Platform interact with the user and with the CLIR Server, it provides to the CLIR Server the query inserted by the user. The CLIR Server take the query and the source and target language and extract the following information:

- Category of the query
- Named Entities (language neutral)
- Ontological Entries (language neutral)
- Common nouns (language dependent)

Then provide a possible translation of all the common nouns recognized based on the source and the target language. These information are send to the Publication Platform that will show the provided translation to allow the user to eventually make correction if the correct focus has not been detected. The translated query can be then used to determine the list of news items that best fit the query.

5.1.1 The CLIR Architecture

This section will describe the overall architecture of the CLIR Server.

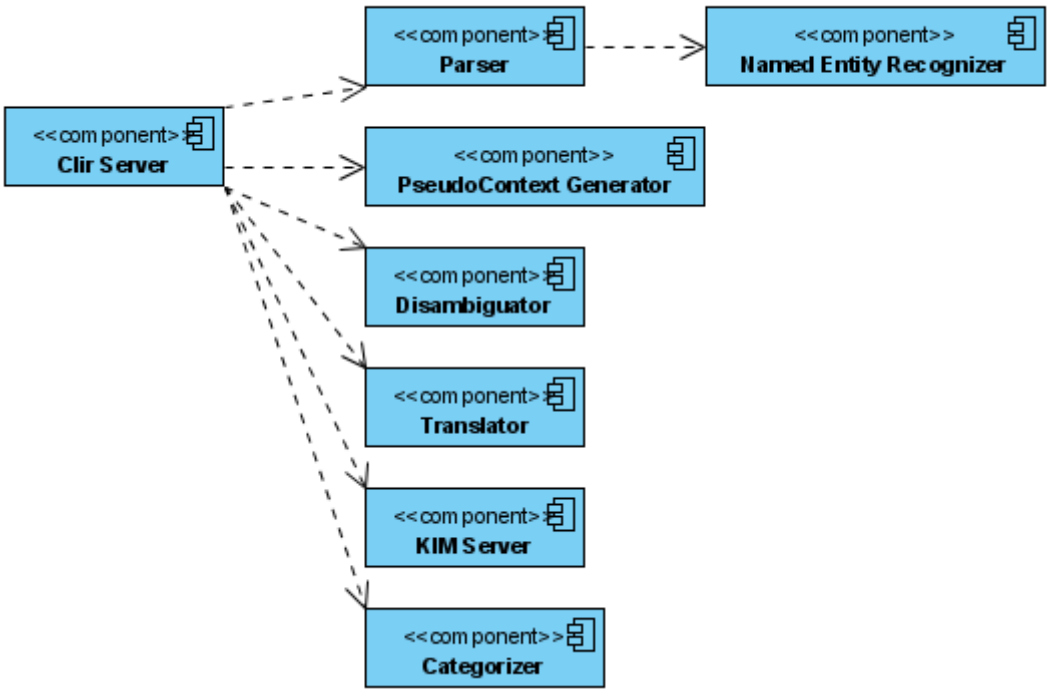


Figure 1

As described in Figure 1, the CLIR Server is composed of several component:

- The Parser, to extract Named Entities and Common Nouns from the query Q;
- Pseudo Context Generator, to generator for a target term T the most relevant terms that co-occur with T;
- Disambiguator, to disambiguate the common nouns in the language L;
- Translator, to translate the common nouns disambiguated from the source language L to the target language L2;
- Kim Server, to annotate with ontological entries the query Q;
- Categorizer to categorize the query Q.

The CLIR Server communicates with these components and managing the internal workflow.

In Figure 2, the overall process is described.

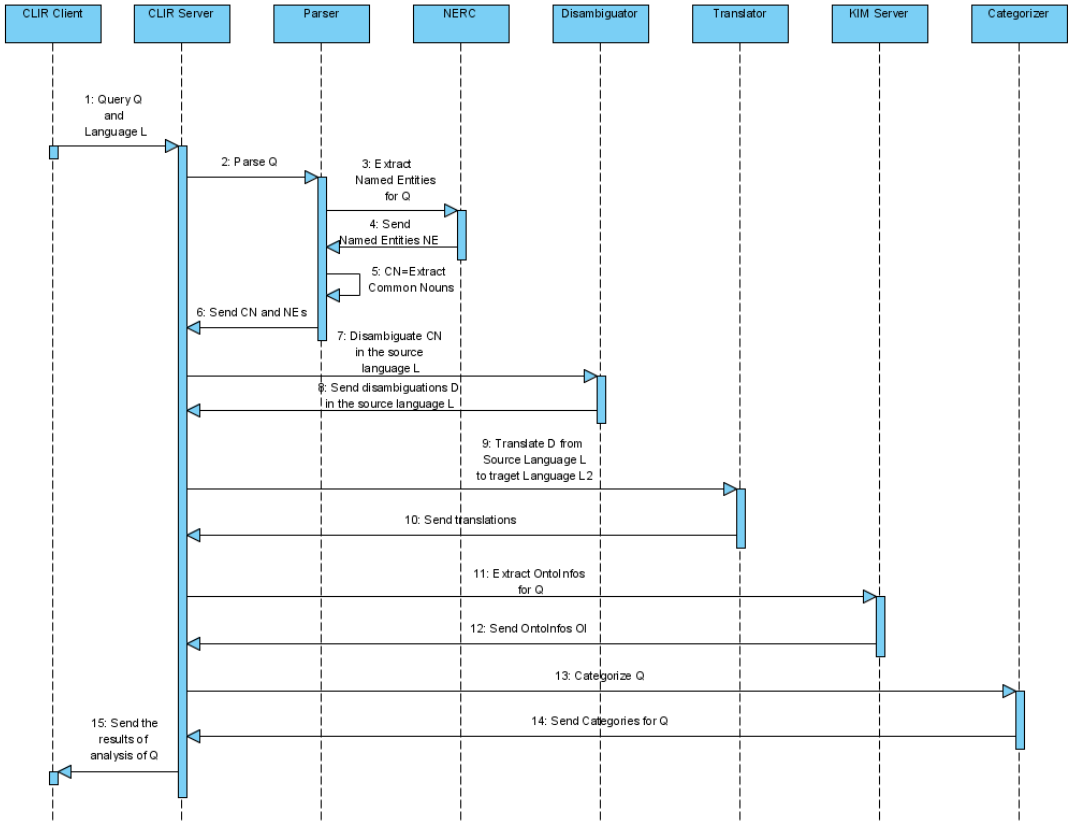


Figure 2

The CLIR Client and Server communicate through a configurable port (socket 5678) and by mean of an XML Stream. The CLIR Client fill the XML with the input query that have to be analysed and the source and the target language, the CLIR Server read these information and according to the languages call the component. As result, the CLIR Server send the output XML to the CLIR Client.

5.1.1.1 Input XML

As an example of the input XML that the CLIR Client has to sent to the CLIR Server, consider the following extract:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<query_session id="0">
<request>
<gui_state>
<user_query>guerra sanguinosa in Irak</user_query>
<gui_language>it</gui_language>
<target_language>en</target_language>
<interaction_level>0.5</interaction_level>
</gui_state>
</request>
<response>
<query_category confidence="0.0">NO CATEGORY</query_category>
<named_entities/>
<ontology_entries/>
<nouns/>
</response>
</query_session>
```

Encapsulated in the tags "user_query", there is the query that the user has inserted and wanted to analyze, in the tags "gui_language" and "target_language" there are respectively the source and the target language. The CLIR Server extract the information from this XML Stream and provide the translations based on the value expressed between the tags "interaction_level". With low values the CLIR Server provides less possible translation, making then more assumption, this implies that more errors can occur. In other words this value, controls the precision of the disambiguation process.

5.1.1.2 Output XML

The CLIR Server, after the elaboration of the XML Stream, responds to the CLIR Client with the following XML Stream:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<query_session id="1">
<request>
<gui_state>
<user_query>guerra sanguinosa in Irak</user_query>
<gui_language>it</gui_language>
<target_language>en</target_language>
<interaction_level>0.5</interaction_level>
</gui_state>
</request>
<response>
<query_category confidence="1.0">
urn:x-prestospace-mad:cs:GenreCS:2005:FOR
</query_category>
<named_entities>
<ne surface="Irak" type="mp7:PlaceType"/>
</named_entities>
<ontology_entries>
<oe surface="Irak" type="Location"
id="http://www.ontotext.com/kim/2005/04/wkb#Country_T.IZ"/>
</ontology_entries>
<nouns>
<noun surface="guerra" lemma="guerra">
<translations best="915394" confidence="0.2">
<sense id="915394" gui_language_lemmas="guerra"
target_language_lemmas="war,warfare"/>
<sense id="896789"
gui_language_lemmas="battaglia,combattimento,conflitto,guerra,lotta,scontro"
target_language_lemmas="battle,conflict,fight,engagement"/>
<sense id="1102789" gui_language_lemmas="guerra"
target_language_lemmas="strife"/>
<sense id="1167074" gui_language_lemmas="guerra"
target_language_lemmas="war,warfare"/>
</translations>
</noun>
<noun surface="sanguinosa" lemma="sanguinosa"/>
<noun surface="Irak" lemma="irak"/>
</nouns>
</response>
</query_session>
```

In this XML Stream, the following metadata are attached:

- the categorisation, between the tags "query_category" with its corresponding "confidence" (plausibility of the category chosen);
- the Named Entities list, between the tags "named_entities";
- for every NE the surface and the MPEG7 Semantic Type is provided;

- the ontological entries list, between the tags "`ontology_entries`";
- for every ontological entry, the surface, the type and the URI is provided;
- the translation proposed for every common noun, between the tags "`nouns`";
- for every translation, the disambiguated sense, the original term and the translation is provided.

6 Study of Applicability to MAD

In this section the applicability of cross-lingual indexing and retrieval tools as discussed in deliverable 16.3 will be studied. This study aims to define the different specific multilingual Information Extraction and Cross-linguistic retrieval tasks as applied for MAD in PrestoSpace. The aim here is to observe the system behaviour on some (preliminary) benchmarking data and to measure the performance of selected tools. In the rest of the project, other data and more involvement of interested users is foreseen and future version of this document will be released.

Examples of the interesting tasks in this phase are the following:

a. Methods for language-driven indexing of multimedia material
b. Methods for information access: browsing, querying and NL querying
c. Management of multilingual information

All the above tasks have been thus selected as candidates for specific steps of performance evaluation of the corresponding tools. To each of the detected tasks will be devoted a specific section in the following.

6.1 Task Ontology-based retrieval (IE)

6.1.1 Description of Task (IE + TC)

The IE task adopted in the MAD (documentation) platform can be defined as follows:

Given a AV data item (program)

- Segment the program into meaningful segments *s*
- For each segment *s*
  - Derive the topical class of *s*
  - Extract NEs from *s*
  - Extract Ontological entities from *s*
  - Extract and lemmatise all the other words/tokens of *s*
  - Compute valid references (i.e. external links) to similar material in external sources (e.g. Web newspapers)
- Upgrade the target data structure (i.e. the EDOB)

The overall information derived from a program is thus:

- A hierarchy of segments
- Metadata related to individual segments: NEs and content tokens (e.g. common nouns and verbs)
- References to the ontology with entities mentioned in each segment
- References to related Web pages

All the above subtasks can be then evaluated properly according to the measures introduced In Section 4.3.2.

6.1.2 IE+TC: Experimental Evaluation

The Rocchio classifier has been learned on a set of manually categorized news (annotated by RAI archivist) includes 1,861 segments of ASR text. A split of 80% for training and 20% for testing was imposed by a random sampling of the data. The 26 RAI categories range from specific classes (like “Economics” and “Foreign Politics”) to more general areas, like “Health”. Each news item was assigned to one or more classes, so that 2,328 assignments were available with an average rate of 1.25 class per news item. News items are not distributed evenly among categories, so that only 11 categories had more than 80 members, i.e. an amount sufficient for a reliable training. Validation was carried out in two fashions:

(1) Measuring the selection of the system among all the 26 classes;

(2) by restricting the testing to only the 11 reliable classes. Results (as BEP points) are reported in Table 1 for the 11 most reliable classes.

The accuracy of the categorizer is satisfactory considering that only a small subset of the archived material from RAI was used for training. In particular small data sets penalize the categories that are more general (i.e. “*Employment/Job*”) although more specific classes require less information to scale up to reasonable accuracy (e.g. “*Sport*”, “*Life and Religion*”). When enough material is available the accuracy confirms the results of benchmarking (e.g. “*Politics*”). Notice how these measures are only based on tokens (BOW, bag-of-word modelling) of the transcribed news and how this material includes a significant amount of noise. Moreover, real-time categorization is ensured by the Rocchio model that is much more efficient[5] than more sophisticated text categorization approaches (e.g. Support Vector Machines).

Category	Training Set Size	BEP (26)	BEP (11)
Sport	76	0,83	0,72
Environment	55	0,45	0,56
Life and Religion	59	0,89	0,79
Current Events	172	0,45	0,54
Economics	149	0,60	0,76
Transportation	48	0,68	0,67
Foreign Affairs	518	0,75	0,78
Justice	346	0,61	0,67
Employment/Job	62	0,55	0,52
Politics	437	0,80	0,79
Health	58	0,73	0,46

Table 1

[5] Profile based classification requires a number of scalar products tight to the number of classes that is much lower to the number of documents.

The validation of the Web alignment capability was carried out on a reference set of about 410 news items (i.e. segments in transcriptions) manually annotated. The annotation was added by a team of three archivists with judged each of the candidate alignment in four classes: "*bad*", "*fair*", "*good*" and "*very good*". The latter expresses an exact correspondence between the event/fact described in the two documents. As the focus of the Web material can be slightly different from the TV news, degrading levels of evaluation express overlaps of decreasing size: "*good*" is a valid correspondence between a transcribed news item shorter than the Web document (which contains many more facts). "*fair*" reflects the same specific topic (e.g. "*Iraki war*") but possibly not the same facts. "*bad*" refers to clear mistakes of the links. In order to study the accuracy of the thresholds, annotators were presented with all the links receiving a score greater than zero[6].

In the evaluation, we wanted to focus on news transcriptions of reasonable quality, i.e. significant segments to accurately measure the linking accuracy. We distinguish between "*monothematic*" and "*multithematic*" units, i.e. segments reporting just one or many more facts, respectively. Multithematic segments are usually due to wrong segmentations that group two or more facts. Annotators found 308 monothematic and 102 multithematic segments. Data reported will refer only to the 1,587 alignments proposed for the monothematic segments.

Two performance indexes were used: *precision* at three levels of *coverage*. *Precision* is the ratio between the number of links that received an evaluation equal or better than the level (from "*fair*" to "*very good*") and the total number of links proposed by the system. Figure 9 plots the three measures according to the thresholds of acceptance imposed to the IR alignment scores. The trends of all the curves suggest that there is a strong correlation between the thresholds and the accuracy. As a contrastive measure, we computed the coverage as the ratio between the number of segments receiving at least one link and the total number of monothematic segments (i.e. 308). In Figure 3, we see that coverage decreases smoothly and a kind of breakeven point is reached in the precision range of 65-75%. This is a quite good result if compared with the standard performance of traditional IR systems. Of course, the constraints imposed on the alignment (in particular, the dates) are quite effective. Moreover, it must be said that not all the segments can be aligned by the system as (1) they may not be present on the "*La Repubblica*" Web site or (2) segments can be too short for significantly express a full fact.

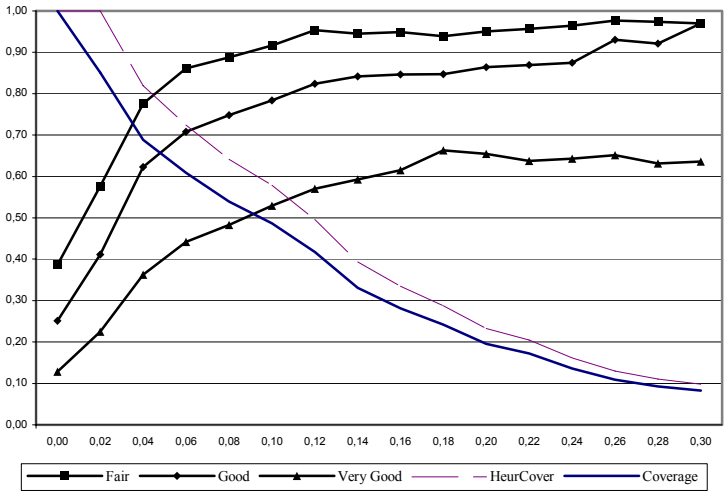


Figure 3

An analysis of the optimal threshold was carried out. We found that, by imposing a threshold of 0.03, the amount of news items not receiving any link was 62 (about 20%). However, we found 48 segments that received only "*bad*" links. An analysis of such 20 segments revealed that there were no Web pages suitable for the alignment on that date (in the adopted source, i.e. "*La Repubblica*"). This because TV news may be local news or curiosity information. Some of them are not even mentioned on newspapers. Correctly, a default threshold of 0.03 would have prevented all the erroneous links to be proposed. Accordingly, we

[6] Notice that recall here does not apply: the annotators did not analyse the full "La Repubblica" Web site in the target time windows so that the gold standard set of all Web news valid for the alignment is not available.

removed those 48 segments from the testing data set (i.e. the 308 monothematic segments) obtaining a reduced set of segments (302-48=254). This simulates a system with a heuristic threshold that correctly assigns no link to the above 48 candidates. Evaluation of such system is focused only on the 254 test segments with an alternative coverage plot ("*Heuristic coverage*" in Figure 3) that is slightly higher than the previous one. Notice how the precision plots for such modified system (by imposing every threshold 0.03 or higher) do not change for any acceptance rate.

In the following tables will be reported, respectively the training and testing data, that has been used to learn and the test the statistical NER for the well written web pages. The document where kept from "La Repubblica" web site, and manually annotated by three annotators.

Class	*Subtype*	*N°*	*Total*
ENAMEX	*Person*	1825	3886
	Organization	769	
	Location	1292	
TIMEX	*Date*	511	613
	Time	102	
NUMEX	*Money*	105	223
	Percent	118	

Table 2 Training material (Web pages)

Class	*Subtype*	*N°*	*Total*
ENAMEX	*Person*	*333*	*537*
	Organization	*129*	
	Location	*75*	
TIMEX	*Date*	*45*	*48*
	Time	*3*	
NUMEX	*Money*	*5*	*13*
	Percent	*8*	

Table 3 Testing material (Web pages)

For the test, 11-fold cross validation (with confidence at 99%) has been applied, and the result are reported in the following table.

	Basic Model	*+Modified Features*	*+Accent treatment*
Average F1	*77.98±2.5*	*79.08±2.5*	*79.75±2.5*

Table 4 Results for the Web pages NER

The statistical NER has also been addressed to annotate text captured by the ASR. In the following tables are reported statistics respectively for the training and testing material.

Class	*Subtype*	*N°*	*Total*
ENAMEX	*Person*	682	1978
	Organization	694	
	Location	602	
TIMEX	*Date*	254	254
	Time	0	
NUMEX	*Money*	102	176
	Percent	74	

Table 5 Training material (ASR)

Class	*Subtype*	*N°*	*Total*
ENAMEX	*Person*	242	702
	Organization	246	
	Location	214	
TIMEX	*Date*	90	90
	Time	0	
NUMEX	*Money*	36	62
	Percent	26	

Table 6 Testing material (ASR)

The performance obtained by the statistical NER over the ASR text are reported in Table 7.

Precision	*Recall*	*F1-measure*
49	56	52.29

Table 7 Results for the ASR NER

The overall performance measured for the Statistical NER applied to the text extracted by the ASR, are lower than the performance obtained applying the same method to well formed sentences of written texts. This can be motivated by the lack of correct syntactic evidence in the ASR text and to the presence of several misspellings, that are not uniformly distributed in the ASR texts. The ASR process in fact can recognize the same speech in different ways according to the environment and the speaker, and homogeneous distribution of the errors are reflected in lower performances.

A specific analysis has been here carried out to better understand the nature and source of the major errors of the ASR NER. An example of transcribed segment is reported hereafter

<DOC>
<DOCNO>1147886605958</DOCNO>
<AN></AN>
<DD> 17/6/2006 </DD>
<SO></SO>
<HL></HL>
<TXT>

e l' uso della mola soluzioni incidente in una frenesia di di <ENAMEX TYPE="LOCATION">via roma</ENAMEX> sono <ENAMEX TYPE="PERSON">maurizio poli</ENAMEX> sono per <ENAMEX TYPE="PERSON">giacomo</ENAMEX> aveva da poco aperto la sua tabaccheria nel quartiere casi

fino a **roma** alle **sette e trenta** come ogni **mattina** poi la tragedia il tentativo di rapina un uomo armato del **clan** il negozio vuole il suo punto di dario

fo si esasperate le ai ripetuti tentativi di furto reagisce viene colpito a morte con un' arma da fuoco al torace e il ladro sarebbe dunque fuggito a piedi lasciando

motorino rubato alcuni giorni fa davanti a taba figlia ci sono le testimonianze che siamo ancora dice si e' visto poco mosca a qualcuno scappare che aveva questo casco da

motociclista le telecamere a circuito chiuso avrebbero potuto riprendere la rapina ma erano fuori uso a tabaccheria subito arrivata la moglie ha avuto un malore e' stata portata in casa

di amici domani sara' effettuata l' autopsia sul corpo del nuovo codice ha ricavato una moglie due figli di cioccolata di sfiducia a pochi metri dal suo negozio su motivi

familiari l' angoscia il dolore ma anche la rabbia dei negozianti dice una la prima volta la legge dove recita e' laureato nevicate che fa per rimanere perche' tutti guardano

mogli ezio con una persona tranquilla piu' persone sara' anche perche' era la persona estremamente buoni addirittura tra i **salvi** dichiara il segreto del mirror che domani **stati uniti** non

era la persona a che gli ha fatto la fed

</TXT>
</DOC>

The comparative analysis of the NER over this segment is reported below:

Document 1147886605958

TAG	TYPE	TEXT	KEY_TYPE	RSP_TYPE	KEY_TEXT	RSP_TEXT
ENAMEX	cor	cor	LOCATION	LOCATION	"via roma"	"via roma"
ENAMEX	mis	mis	PERSON		"maurizio poli"	""
ENAMEX	mis	mis	PERSON		"giacomo"	""
ENAMEX	spu	spu		LOCATION	""	"roma"
ENAMEX	spu	spu		ORGANIZATION	""	"clan"
ENAMEX	spu	spu		PERSON	""	"salvi"
ENAMEX	spu	spu		LOCATION	""	"stati uniti"
TIMEX	spu	spu		TIME	""	"sette e trenta"
TIMEX	spu	spu		DATE	""	"domani"

Here we can see that the system correctly recognizes one NE (first row) where the correct NE type (KEY_TYPE) and the correct text (KEY_TEXT) fully overlaps with the response type (RSP_TYPE) and text (RSP_TEXT). There are two correct NE second and third rows) that are not recognized by the ASR NER and 6 wrongly introduced NEs (last six rows denoting spurious entries). The performance here is very low as the above table implies.

However some observations apply. First the two missing entries “maurizio poli” and “giacomo” although are credibly persons appear into a context where it is really very hard to validate them, even manually. The context here is like

“*sono maurizio poli sono per giacomo aveva …*”

(literally) “(they) *are maurizio poli* (they) *are for giacomo* (he) *had* …”

and it is by no way clear. It is not only syntactically illegal, but even meaningless to any reader. Although the human annotator gave us the two name entities, without access to the original spoken report, it is not possible to assess the annotation and establish if the oracle is correct. The missing information of the ASR NER is likely due to the odd context that does not support (as it stands) any conclusive decision.

Finally, the spurious NE introduced by the ASR NER are reasonable choices like “*roma*” as a location, “*clan*” as an organization and one time expression in the sentence:

fino a **roma** alle **sette e trenta** come ogni mattina poi la tragedia il tentativo di rapina un uomo armato del **clan** il negozio vuole il suo punto di dario up to **rome** at **seven thirty** like every morning then the tragedy the attempt of a robbery an armed man of the **clan** the shop (he) wants his point of the dario

It must be said that although “*clan*” is not a proper noun in Italian, it is usually adopted to refer to a specific (previously mentioned) criminal organization (e.g. *mafia*). On the other side, all the others proposed NEs (like “*roma*”) are valid and the context does not fully clarify if they are correct or not. They were simply neglected by the annotator that (like the system) cannot access to the original speech.

It seems that the recognized NEs arise from typical contexts and the more likely they are in the newspaper prose, the higher is the confidence of the system (e.g. “*roma*” as a location). It is of course a heuristic method but seems to work well. For example, the previous mistakes about the “*maurizio poli*” refers to a PERSON that is likely to be unknown to most people; the other “*giacomo*” (a First Name in Italian) cannot be linked to any specific person.

Finally, in the following sentence

mogli ezio con una persona tranquilla piu' persone sara' anche perche' era la persona estremamente buoni addirittura tra i **salvi** dichiara il segreto del mirror che domani **stati uniti** non wifes ezio with a calm person more persons (it) will also because (he) was the person extremely nice indeed among the salvi (he) declares the secret of the mirror that tomorrow ***United States*** not …

stati uniti (*USA*) is recognized although it seems unlikely that it was a proper transcription. However, as that the only evidence is given by the transcribed text and *stati uniti* is very frequent in newspapers and mostly unambiguous in Italian, the system outputs it anyhow. The amount of knowledge necessary to neglect such a name is definitively too big to avoid such errors.

A quick conclusion about these data is the fact that evaluation of the ASR NER is very complex as it is sensible to errors of the ASR module. The ASR NER assumes as the only evidence the recognized context so that performance figures cannot be obtained according to a “theoretical correctness” given by an informed tagging. Some of the human annotator choices in fact can be refused: it is in fact not clear if “*roma*” as a location was wrong in the second discussed sentence. Without access to speech we are forced to live with some assumptions based on the likelihood of some names (e.g. “*roma*” or “*stati uniti*”).

Table 7 is thus reporting the cumulative error of the ASR and the ASR NER and this is too restrictive. A second tagging has been thus carried out over the test material, by relaxing some of the former criteria. For

example, cases like "*stati uniti*" were assumed as valid NE although their transcription could be debatable. In this way we tried not to take into account the original speech but the same textual context available after ASR. For example, all the NE ("roma"/LOCATION, "sette e trenta"/TIME) in the sentence "*…fino a **roma** alle **sette e trenta** come ogni mattina …*") were considered valid NEs according to the coherent context made available. This new measure tends to factor out the ASR errors during the test of the ASR NER. The new performance measures have been obtained without retraining the system and the results are summarized in Table 8.

Precision	*Recall*	*F1-measure*
89.90	57.60	70.23

Table 8 Results for the ASR NER

The results suggest that without considering the errors originated by the transcriptions, the system is highly more precise: the F2-measure, given by doubling the weight given to precision, i.e. $\frac{3pr}{2p+r}$, equals 80.84%.

It is to be said that at this stage of the project the ASR NER module has been partially trained as only a limited data set has been manually annotated. In the remaining project activities, a consistent revision of the learning process is foreseen. Larger training data sets will be made available to improve the parameter settings. Moreover, a stricter tagging policy will be studied through the integration of the HMM model with the system internal vocabularies (i.e. gazetteers) in order to increase the overall precision. These two lines of exploration (more extensive annotation and fine tuning) will allow to achieve an effective automatic NE tagging also on ASR transcriptions.

6.2 Task Ontology-based retrieval (OR)

Description of Task (OR)

The ontology based retrieval process can be schematised as follows:

Given a user need *c* (i.e. a concept)

- Locate the ontological class *C* of *c*
- Navigate through the ontology from *C* and for each reachable concept *C'*
  - Follows potential links to documents (i.e. EDOB segments) related to C'

The ontology based retrieval can be viewed as an extension of the traditional keyword retrieval known by the contemporary search mechanisms. The advantages of the former are based in the extended capabilities for specifying the user need by providing a semantic description of the resources in demand.

The complexity of this search needs not to be imposed on the user of the system and various approaches include natural or controlled language queries, or intuitive user interfaces like faceted search and others. On the other hand the complexity of the pre-processing of the resources and the evaluation of the search criteria is significant. The pre-processing needed for the traditional keyword based search can be summarized as follows:

- Identify tokens/words in the textual representation of the content;
- Generate a full-text search index over these tokens: basically the link between each sequence of characters and the resources which contain it;

In the situation where metadata fields are used to further describe the resource the same technique applies for each of these fields.

6.2.1.1 Preparation

The advantages of ontology based search, although significant, pose a challenge in terms of needed preparation of a domain model, pre-processing of the resources and evaluation of the semantic queries. The preparation of the model consists of two parts:

- Ontology schema preparation: The ontology schema is an abstract model of the domain or the "world" in focus for the search criteria. Usually this model covers the concepts and relations between them which will be useful for the intended search applications. This way the model's complexity can be optimised to what is needed and prevent one from falling into deeper philosophical questions about the concepts and their properties;
- Ontology (pre-) population of instances: In many cases algorithms for Information Extraction benefit from previously known facts about the content they are about to process. Examples of such facts may be lists of first names of people or geographical denominations. Although the model is already in place the gathering of the "population" of the model is a process of higher scale and is best performed in semi-automatic manner from trusted sources (e.g. geographical gazetteers, yellow pages, etc.);

After having the model, the pre-processing of each resource that needs to be findable should be done in the following way:

- Distinguish tokens;
- Recognise entities (quite a complex task usually involving IE or heavier techniques);
- Identify entities: i.e. find if they exist in the ontology or should be added as new. Also quite a complex tasks similar to record linkage in databases and sometimes called Identity Resolution;
- Build FTS index;
- Build Semantic Search Index (i.e. having a link between the entities and the resources in which they appear);

6.2.1.2 Query Evaluation

There are two main aspects of ontology based search which differ in the expressivity of the need definition and slightly in the complexity of the query evaluation. The first of them called "Entity Lookup" allows for specification of the class of the instance and (a part of) its name.

Example: "Give me people which name contains Mac"

The evaluation of this query, once the class of the demanded entities is identified (Person) should go like this:

- Find all the subclasses of the specified class (e.g. Man, Woman subclasses of Person);
- Find all the entities in this set of classes which meet the name restriction;
- Follow potential links to documents (i.e. EDOB segments) related to one of the entities in the result set;

Note: Technically this schema is not applied literally for reasons of performance optimisation. Instead, all the restrictions are applied in a single query leading to immediate narrowing down of the result candidates.

The other aspect of the ontology based search is evident when the user need specifies a more complex pattern of relationships and class-affiliations.

Example: Give me all people having a political position in a geographic location named Bulgaria.

Such a definition includes the specification of a pattern in the ontology (similar to a path in a directed graph) built by entity lookups (class and name restriction) linked to each other through relationships/properties. An extension to this is the specification of attribute restrictions (Person of age 16). The difference here is that the attribute has a literal value and is not a fully operational instance having independent existence outside the context of the owner entity. Finally one can choose which entity restrictions in the pattern should be included in the result set, while the others will be used just as a specification of the demand.

Example: considering the last example, I am interested just in the people and their position, and not in the geographic location.

The evaluation of the pattern search can be schematised as follows:

- Do entity lookup for each of nodes of the pattern;
- Filter out the entities by the relationship restrictions between the nodes;
- Return the entities from the demanded nodes using the others just as a specification of the restriction;

Note: Again, the actual evaluation tries to minimise the intermediate results by following the most restrictive parts of the query and the naïve schema provided here should not be implemented as described.

6.3 Task Cross-linguistic Retrieval (CLIR)

6.3.1 Description of Task (CLIR)

The CLIR task can be schematised as follows:

a. Given a query *q* in a source language L1
   i. Derive the topical class of *q*
   ii. Extract NEs from *q*
   iii. Locate Ontological entities in *q* from L1
   iv. Extract and lemmatise the source question *q* in L1
   v. Locate the expanded query (i.e. NEs, Ontological indexes, nouns in L1) into the LSA space
   vi. Disambiguate each noun of *q* in L1 via domain-driven disambiguation in WordNet
   vii. For each disambiguated noun n
      1. Select the WordNet synset of *n* in the target language L2
      2. Substitute each original noun *n* in L1 with the corresponding synonyms *s(n)* of L2

b. Append to the derived noun set $SN=\cup_n s(n)$, the NEs *NE(q)* of q and the translations *OL2(q)* in L2 of all the ontological items detected in the source query q

c. Retrieve all EDOS segments via the expanded query: $SN \cup NE(q) \cup OL2(q)$

6.3.2 CLIR: Experimental Evaluation

The quantitative evaluation of the CLIR task would require a significant set of queries and a manually compiled corpus of document considered relevant for them. At the current stage of the project this material to be used as an oracle (See section 4.3.2) is not available. On the contrary a qualitative analysis will be reported here. A set of complex queries are considered and their cross-language application is studied. In particular the query expansion technique discussed in Sect. 3.3.2 will be here targeted: the translation computed by the algorithm of the previous section will be analysed and its impact evaluated. The accuracy of the CLIR queries will be manually verified according to the quality (correctness) of individual translations. The following tables report the analysis of a set of 8 queries.

Examples of CLIR from English (query) to Italian (retrieved segments)

<table>
<tr><td>Input Query</td><td colspan="3">Blair calls on NATO member to contribute more troops to Afghanistan force.</td></tr>
<tr><td>Chaos NEs</td><td colspan="3">Blair [person]
NATO [organisation]
Afghanistan [paese]</td></tr>
<tr><td>KIM NEs</td><td colspan="3">NATO [Organization]
Blair [Person]
Afghanistan [Location]</td></tr>
<tr><td rowspan="8">Nouns, Translations</td><td>Noun</td><td>Input language senses</td><td>Target language senses</td></tr>
<tr><td>NATO</td><td>North_Atlantic_Treaty_Organization, NATO</td><td>n.a.t.o., organizzazione_del_trattato_nordatlantico</td></tr>
<tr><td rowspan="5">member</td><td>member</td><td>componente, membro</td></tr>
<tr><td>penis, phallus, member</td><td>asta, fallo, membro, membro_virile, pene, verga</td></tr>
<tr><td>member</td><td>appartenente, componente, iscritto, membro</td></tr>
<tr><td>extremity, appendage, member</td><td>arto, estremita', membro</td></tr>
<tr><td>member</td><td>membro</td></tr>
<tr><td>troops</td><td>military_personnel, soldiery, troops</td><td>truppa</td></tr>
<tr><td rowspan="6">force</td><td>force</td><td>forza</td></tr>
<tr><td>military_unit, military_force, military_group, force</td><td>arma</td></tr>
<tr><td>violence, force</td><td>forza, violenza</td></tr>
<tr><td>effect, force</td><td>effetto, forza</td></tr>
<tr><td>force, personne</td><td>forza, personale</td></tr>
<tr><td>force, forcefulness, strength</td><td>corpo, energia, forza, lena</td></tr>
<tr><td>Afghanistan</td><td>Afghanistan, Islamic_State_of_Afghanistan</td><td>afghanistan</td></tr>
<tr><td>Output Query</td><td colspan="3">Person:Blair & Organization:Nato & Location:Afghanistan & (n.a.t.o | “organizzazione del trattato nordatlantico”) & membro & truppa & arma</td></tr>
</table>

Input Query	Guardian Unlimited news on your mobile - free trial.		
Chaos NEs	Guardian [No category available] Unlimited [No category available]		
KIM NEs	Guardian [Object]		
Nouns, Translations	**Noun**	**Input language senses**	**Target language senses**
	news	news, intelligence, tidings, word	notizia, novella
		news	**informazione, notizia, nuova**
		news	annuncio, annunzio, cronaca, informazione
	trial	trial	giudizio, sentenza, verdetto
		test, trial	cimento, prova
		test, trial, run	cimento, esperimento, test]
		trial, tribulation, visitation	tribolamento, tribolazione
		trial, trial_run, test, tryout	prova
		trial	**giudizio, processo**
Output Query	Object:Guardian & (informazione | notizia) & (giudizio | processo)		

<table>
<tr><td>Input Query</td><td colspan="3">A series of bomb attacks on London's metro network.</td></tr>
<tr><td>Chaos NEs</td><td colspan="3">London [citta]</td></tr>
<tr><td>KIM NEs</td><td colspan="3">London [Location]</td></tr>
<tr><td>Nouns, Translations</td><td>Noun</td><td>Input language senses</td><td>Target language senses</td></tr>
<tr><td></td><td>series</td><td>series
serial, series
series</td><td>catena, ordine, serie
sceneggiato, serial
serie</td></tr>
<tr><td></td><td>bomb</td><td>bomb</td><td>bomba</td></tr>
<tr><td></td><td>London</td><td>London, Greater_London, British_capital, capital_of_the_United_Kingdom</td><td>londra</td></tr>
<tr><td></td><td>metro</td><td>metro, subway, tube, underground</td><td>metro, metropolitana</td></tr>
<tr><td></td><td>network</td><td>network
net, network, mesh, meshing, meshwork]</td><td>rete_delle_comunicazioni
maglia, rete, rete</td></tr>
<tr><td>Output Query</td><td colspan="3">Location:London & (catena | ordine | serie) & bomba & (metro | metropolitana) & (maglia | rete)</td></tr>
</table>

<table>
<tr><td>Input Query</td><td colspan="3">Minister Straw, London bombing and the hallmark of an Al-Qaeda attack.</td></tr>
<tr><td>Chaos NEs</td><td colspan="3">Minister [persona]
Straw [No category available]
London [citta]
Al-Qaeda [No category available]</td></tr>
<tr><td>KIM NEs</td><td colspan="3">Straw [Person]
London [Location]
Al-Qaeda [Organization]</td></tr>
<tr><td>Nouns, Translations</td><td>Noun</td><td>Input language senses</td><td>Target language senses</td></tr>
<tr><td></td><td>bombing</td><td>bombing, bombardment</td><td>bombardamento</td></tr>
<tr><td></td><td>London</td><td>London, Greater_London, British_capital, capital_of_the_United_Kingdom</td><td>londra</td></tr>
<tr><td></td><td>hallmark</td><td>hallmark, trademark, earmark, stylemark</td><td>cachet, marchio</td></tr>
<tr><td></td><td>attack</td><td>attack
attack, onslaught, onset, onrush
approach, attack, plan_of_attack
fire, attack, flak, flack, blast
attack, attempt
attack, tone-beginning</td><td>accesso, crisi
assalto, attacco
approccio
attacco
attentato
assalto, attacco</td></tr>
<tr><td>Output Query</td><td colspan="3">Person:Minister & Person:Straw & Location:London & Organization:Al-Qaeda & bombardamento & (chachet | marchio) & attacco</td></tr>
</table>

Examples of CLIR from Italian (query) to English (retrieved segments)

<table>
<tr><td>Input Query</td><td colspan="3">Berlusconi al parlamento sulla missione di guerra in Iraq.</td></tr>
<tr><td>Chaos NEs</td><td colspan="3">Berlusconi [person]
Iraq [paese]</td></tr>
<tr><td>KIM NEs</td><td colspan="3">Iraq [Location]</td></tr>
<tr><td rowspan="5">Nouns, Translations</td><td>Noun</td><td>Input language senses</td><td>Target language senses</td></tr>
<tr><td>parlamento</td><td>parlamento</td><td>parliament</td></tr>
<tr><td>missione</td><td>delegazione, deputazione, missione, rappresentanza
missione</td><td>deputation, commission, delegation, delegacy, mission
mission, military_mission</td></tr>
<tr><td rowspan="2">guerra</td><td>guerra
battaglia, combattimento, conflitto, guerra, lotta, scontro
discordia, disunione, guerra, zizzania</td><td>war, warfare
battle, conflict, fight, engagement
discord, strife</td></tr>
<tr><td>guerra</td><td>strife</td></tr>
<tr><td>Output Query</td><td colspan="3">Person:Berlusconi & Location:Iraq & parliament & (deputation | commission | delegation | delegacy | mission) & strife</td></tr>
</table>

Input Query	Ritorno dell'esercito italiano dalla missione NATO in Afghanistan.		
Chaos NEs	NATO [company] Afghanistan [paese]		
KIM NEs	Afghanistan [Location] NATO [Organization]		
Nouns, Translations	**Noun**	**Input language senses**	**Target language senses**
	missione	delegazione, deputazione, missione, rappresentanza	deputation, commission, delegation, delegacy, mission
		missione	**mission, military_mission**
	esercito	caterva, esercito, falange, legione	horde, host, legion
		esercito, forze_armate	military, armed_forces, armed_services, military_machine, war_machine
		esercito	army
		armata, esercito, falange, milizia	**army, regular_army, ground_forces**
Output Query	Organisation: NATO & Location:Afghanistan & (mission | “military mission”) & (army | “regular army” | “ground forces”)		

Other synthetic results are reported below:

Input query	Chaos NEs	KIM NEs	Nouns, Translations
Middle East minister comments the cease-fire and the UN resolution.	UN [No category available]	UN [Organization] Middle East [Location]	minister **[ministro]** cease-fire **[armistizio, cessate_il_fuoco, tregua]** UN **[n.u., nazioni_unite, o.n.u., organizzazione_delle_nazioni_unite]** resolution **[risoluzione]**
Howells admits doubts over Lebanon.	Howells [No category available] Lebanon [paese]	Lebanon [Location]	Lebanon **[libano]**
NATO fails on Afghan troops plea.	NATO [organisation]	NATO [Organization]	NATO **[n.a.t.o., organizzazione_del_trattato_nordatlantico]** troops **[truppa]** plea **[implorazione, preghiera, supplica]**

Input query	Chaos NEs	KIM NEs	Nouns, Translations
Dibattito nella maggioranza in conflitto: delibera sulla missione in Libano.	Libano [No category available]		maggioranza **[majority, bulk]** conflitto **[clash, friction]** delibera **[decision, determination, conclusion]** missione **[deputation, commission, delegation, delegacy, mission]**
Dibattito nella maggioranza in conflitto: delibera sulla missione in Iraq.	Iraq [paese]	Iraq [Location]	maggioranza **[majority, bulk]** conflitto **[clash, friction]** delibera **[decision, determination, conclusion]** missione **[deputation, commission, delegation, delegacy, mission]**

From the above examples it is possible to draw an initial although very partial performance analysis.

The evaluated English-to-Italian (E2I) queries are 8 while the Italian-to-English (I2E) queries are 4. The target decision here is the translation of individual senses of nouns in the queries. Individual decisions are 28 nouns for the E2I queries and 13 nouns for the I2E ones. Accidentally, the number of senses per noun (i.e. the average ambiguity of the nouns in the queries) was 2.6 for both the E2I and I2E nouns. Notice that for this task (differently from ad hoc IR tasks) recall is equal to precision as each individual decision has exactly one positive sense and we will call it *accuracy* (i.e. 1-error rate). The accuracy is about 0.87% per E2I queries and 0.84% for the I2E ones. When a sense is correctly assigned then we assume that the translations proposed are also correct even if some lack in the resource (i.e. the alignment between the source and target language at the sense level) may produce unsatisfactory translated nouns. In synthesis, from the test we can assume that all the queries are satisfactorily translated in 87% and 84% of the analyzed cases. Notice how the synonymy in the target language (i.e. the fact that more than one Italian/English noun is activated by the translation of an individual English/Italian noun) provides that translated queries are even richer than the original queries, so that the overall achievable recall is also boosted.

The very good results, even on this limited data set, suggest that the direction of this methodology is more than promising and its applicability very large. However, more measures are required to assess the quality of the translation in general, i.e. to evaluate the *internal* ability of the model to produce accurate translations in the target language. It is to be also noticed that more concrete results will be available when the proposed CLIR method will be evaluated *externally* against the final target task, i.e. multilingual retrieval. In this case the comparison of the precision and recall scores of the retrieval process, with or without translations, will provide an indirect, but more useful, indication about the effectiveness of the CLIR model introduced here.

7 Technological Perspectives in PrestoSpace

7.1 Global Analysis of results

The above set of experiments gave rise to a rather comprehensive analysis and is already a confirm of the potentials of the overall MAD to semantic indexing and retrieval. Although user oriented measures have not yet been applied (as this is part of the remaining phases of the PrestoSpace project) most in vitro experiments over real data sets are successful. Some weaknesses will be discussed in the next section and related technologies not yet directly adopted in MAD will be also suggested.

The experiments have been often applied on individual stages of the MAD processes. For example categorization has been tested on somewhat ad hoc data. The training and testing data sets were derived in fact from a manually segmented editorial parts, that were thus error free. This is a strong abstraction if compared with the fully automatic nature of the MAD semantic analysis process. On one side, we have thus been able to measure the effective individual contribution of the text categorization module. On the other hand we could not directly observe the errors due to the influence of noisy data as they are derived by the preceding automatic content analysis. Finally, no data is still available regarding the user perceived accuracy that is usually higher than precision/recall objective measures.

A second issue is the relationship between the automatic production of semantic metadata and the manual documentation process. Most of the errors due to dangerous dependencies between MAD processes (i.e. producers and consumers of relevant information like speech transcriptions and named entity recognition) can be removed by the manual documentation for which specific interfaces have been designed. When more evidence about performances and documentation effort will be available (user testing is on going at the time of writing this document) further architectures for documentation can be foreseen. For example, manual validation of partial information (e.g. editorial segmentation) can be applied before highly dependent processes (e.g. Named Entity recognition and Text Categorization) are applied. In this case cascades of errors that cumulate negatively across the MAD pipeline of individual GAMPs can be avoided. An interesting more dynamic architecture can be foreseen in which, during manual documentation, some

GAMPs are automatically rerun in order to dynamically take into account the latest manual corrections. This would exploit correlations measured during this preliminary evaluation phase and suggest proper and more cost-effective documentation models.

A final remark is on the adopted objective measures adopted for quantitative evaluation, i.e. precision recall factors. It has been often argued that they are not fully capturing the nature of the retrieval problem and this is even more valid for documentation. Part of the evaluation not yet available at this stage of the project should involved user-dependent factors like usability of the interface and mean-time to information in real settings. Part of the extensions of this document (that can be considered after this first issue as still living) will include discussion and measures on this line.

7.2 Current limitations

The tests carried out suggested a number of limitations that the current technology, even if close to state-of-the-art for most of its components, implies within the documentation and publication processes foreseen by the MAD PrestoSpace objectives.

First of all is the segmentation quality. As most of the semantic analysis is carried out on a unit of information consisting of an individual editorial segment, errors in the detection of the proper boundaries of these segments irreparably propagate to semantic metadata. An example is the case where more than one segment in a broadcasted TV program are mistakenly merged into one. In this case, the lexical evidence brought as input to the text categorization of the SA_GAMPs is misleading as it suggests in general more than one valid class: this produces lower scores for the potentially correct classes in the best case. Sometimes it produces a o answer situation when words about different topics (A and B, for example) cannot cumulate enough evidence to trigger any of the involved classes (neither A nor B are detected). The current markovian model for Named Entity recognition is also in the class of modules that are sensible to segmentation quality. In this case, the recognition and boundary detection for individual Named Entities are computed from the most likely chains from the observable word sequences. Erroneous segmentations, that mix sentences of different nature (for example appending “sport” statements to “politics” prose) causes great variations of the involved probabilities with a significant impact on the output quality. Notice how, at the moment, independent segmentation algorithms are applied at lexical, video and audio levels and their integration is based on a simple combination (i.e. voting) strategy. Additional study here is possible about the adoption of more complex integration strategies that make a deeper use of the different available evidences, like visual, i.e. image and video based, and conceptual, i.e. lexically and textually driven, information. A discussion on this aspect is reported in the next section.

A further set of problems in the MAD semantic analysis is the limited accuracy currently reached by some GAMPs. An example is the accuracy of the Named Entity Recognition and Classification (NERC) component of the Italian Semantic Analyser. Although most of the problems are here due to the noisy nature of the Automatic Speech Recognition (ASR) technology that do not allows to apply NERC algorithm natively inspired by written text analysis to be effective. ASR transcripts are in fact not governed by the normal rules of language grammaticality and are also not capitalized. The result is a pseudo-text whose sequences do not provide systematic contextual evidence to easily detect the boundaries of complex named entities (e.g. in “the President of the European parliament”). These weak contexts are at the basis of the inherent complexity of the NERC task, given that most proper nouns (e.g. person proper nouns like “Prodi”) are also common nouns. It is certainly true the improvement in the ASR technology in the future will keep these undesirable effects limited. However, some work can still to be done to improve the actual quality of the NERC process given the currently available ASR tools. The adopted solution to improve the robustness of the NERC component is now based on Hidden Markov models (HMM), as they can exhibit simple training procedures and efficient recognition algorithms. The current work in PrestoSpace is concentrating in augmenting the set of available of manually annotated text transcriptions as they can be importantly used to improve the training of the applied HMM technology.

A further set of limitations in the technology are certainly due to architectural choices. Most of the GAMPs are in fact working at a single media level, i.e. audio, video or ASR transcripts. The actual workflow is a cascade of different processes and the only dependences are given by input requirements: the SA GAMPs for example require ASR transcripts but make no use of video or image information/properties. Due to the inherent multimedia nature of much of the involved information some deeper integration among modules would be beneficial. A better processing strategy should be tested in the future according to more flexible architectures and GAMPs, where multiple runs should be allowed in order to revise some early choices

(e.g. video segments) whenever more evidence (e.g. lexical information) is made available by later GAMPs (e.g. the ASR GAMP). Of course this extensions will have an impact on several design choices of the project, these including the unified data model (EDOB) designed. As most technologies have been studied at individual media levels, the MAD architecture, as it is now, is a perfect framework where design, experimentation and assessment of more multimedia oriented SA algorithms or tools can be now on carried out. As it stands, the MAD platform, with its merits and limitations, constitutes one of the rare complete infrastructures where this applications and lines of research are enabled.

7.3 Potential Extensions and Related Technologies

The previous discussion outlined a number of weak points of the current semantic analysis and cross-linguistic approaches proposed by the MAD component of PrestoSpace. As these lines represent work in progress at the research as well as at the technological level, an overview on some perspective work along them can be done at this stage of the project.

The integration of different editorial segmentation algorithms is certainly a relevant issue. An approach based on an Hidden Markov models can be here presented and is discussed in section 7.3.1. Although the impact of such approach on the current data model is not discussed in this document, it is to be taken into account that a every complex interaction among GAMPs reduces the overall modularity of the MAD current approach and reflects on some choices made on the EDOB XML infra-structure. The design of a new unified segmentation processor has this as a consequence a non trivial revision of the EDOB data model.

An important extension to be explored in the near future is the enrichment of the types of semantic metadata extracted by the current IE technology adopted in PrestoSpace. Recent advances in the detection of relevant semantic phenomena involve the rather new field of shallow semantic parsing also referred to as semantic role labelling. The aim of these technology is to automatically capture in raw text the major relationships between the sentence main entities. This represents a form of relation extraction that is usually carried out according to state-of-art adaptive methods (kernel based machine learning and support vector machines). A survey of this technology as a result of the extensive research carried out in PrestoSpace is reported in section 7.3.2.

Another aspect related to future extensions is the relationship between the MAD data model and some currently explored digital library standards for Web IR. A relevant case to look at is the SRW/SRU standard [Morgan:2004], [Sanderson et al.:2005]. SRU (Search/Retrieve via URL) is a standard search protocol for Internet search queries, utilizing CQL (Common Query Language), a standard query syntax for representing queries. SRW (Search Retrieve Web Service) is a companion protocol to SRU. The Library of Congress serves as the maintenance agency for these standards. SRU/SRW are standards that attempt to integrate the notions of Information Retrieval Web Services and Digital Libraries. This is of particular relevance for the Documentation and Publication/Search processes in PrestoSpace. Other framework to look for in the specific area of radio and TV news is the RSS publication standard adopted by most news agencies and Web journals. Notice how these infrastructures tend to map the publication stage into a pushing processes. A significant exploitation of this technology can be made internally to individual GAMPs. The Web spidering process foreseen by the SA GAMPs is actually based on a spidering tailored to specific sites (e.g. on-line newspapers). By exploiting the RSS technology the GAMPs can be significantly simplified as it has local access to all published material from trusted sources according to a standard and portable interface. This is of course a simple but important extension that will be part of further exploration in the near future.

7.3.1 Markov models for editorial segmentation

The segmentation problem is an critical issue for the quality and accuracy of most of the Information Extraction tools related to semantic information. Currently, the technology is relatively well assessed for TV news as editorial segment detection is driven by video, audio and lexical features. A combination of them is adopted as a kind of weighted voting approach.

An adaptive approach could be applied whenever a sufficiently large number of documented EDOB material will be made available by PrestoSpace users. The segmentation problem in fact depends on a

chain of events that are expressed cooperatively by the different media levels: change of scene, change of the speaker and/or change of topics. By treating all this evidence an a temporal basis (i.e. as a discrete set of time stamps where the end of the current segment may or may not happen) and assign them features to the chain of individual time points, the segmentation process can be mapped into the state recognition problem of an hidden information source and solved via Bayesian probability methods characterizing Hidden Markov Models (HMM, [Jelinek:1993]). The advantage of this approach would be to be self adaptive to different audiovisual material related to different types of programs (from TV news to documentaries of different nature). In fact, the modelling would be preserved while the different training would be made available by the early activities of documentation expert that would provide the correct segmentation by hand for a limited number of AV items: it would suffice for reaching a good accuracy soon in the porting of the PrestoSpace application towards a new domain. A discussion of a possible HMM-based approach to editorial segmentation can be found in Appendix 1.

7.3.2 *State-of-art* Information Extraction through SVMs and kernel methods

Semantic analysis of natural languages can be performed at different levels of granularity: single words can be associated with the concepts they refer to; word sequences, phrases and clauses can be related to each other; the meaning of a discourse or a dialogue as a whole can be investigated. This latter is called a Natural Language Understanding (NLU) problem, and is sometimes referred to as an AI-complete problem by analogy to NP-completeness in complexity theory. In fact, some problems that should be addressed by a NLU system, such as anaphora resolution or the handling of quantifiers in logic inference, require so many linguistic, ontological, pragmatic and contextual information that it seems unlikely that they can be successfully addressed and resolved in an open domain fashion.

A softer approach to semantic analysis consists in restraining the scope of the problem, limiting the study to the development of models that establish semantic relations between words within a sentence, or between groups of words in a sentence and their linguistic context. This kind of analysis is called Shallow Semantic Parsing, the adjective *shallow* stressing the fact that the models at study do not pretend to capture the whole semantics of a sentence, or document, or discourse as a whole, whereas they rather focus on some very precise aspects of the semantics of natural language texts, assuming that such aspects:

- can be represented with appropriate (and measurable) accuracy;
- are interesting for the specific application domain.

Semantic Role Labelling (SRL) is an approach to shallow semantic parsing which is especially interesting for many common information retrieval and natural language processing tasks, such as event extraction, question answering and classification, document and passage classification and retrieval, ontology learning and so on.

The task consists in the identification of predicate argument structures – or propositions - within natural language sentences and texts. A proposition consists of a predicate, i.e. the part of the sentence that shapes an assertion about a subject, along with the word sequences – the predicate arguments – that complete the meaning of the assertion.

Predicate argument structures are a powerful instrument for information retrieval, information extraction and data mining systems, as the information they encode presents many major advantages over purely lexical approaches such as bag of word models. In fact, a predicate argument structure:

- provides a very high level of abstraction, as it generalizes the informative content of many different lexicalisations of the same concept;
- allows to tune easily a search by relaxing or reinforcing semantic restrictions enforced on the argument slots of each predicate;

- allows for cross-language reasoning, as it constitutes a language independent representation of situations and events. This requires (1) a translation of the proper meaning of the predicate lexicalisations and (2) an appropriate mapping between the world models describing the different corpora.

At the highest level of abstraction, a semantic role labelling system is a black box that receives some free text as input and outputs a semantic annotation of the input text. The adjective *free* in *free text* means that it doesn't contain any meta-information or special formatting, it is just a string of text as a user would write for other persons to understand it. The output annotation is a representation of the input sentence in which the semantic structures that have been identified are somehow marked. The task is accomplished using some external resources, such as knowledge bases, linguistic tools and, most notably, statistical models.

In fact, the SRL process is typically modelled as a combination of statistical machine learning problems. This approach is motivated by several reasons, mainly the existence (1) of very accurate and precise supervised learning algorithms and (2) of large corpora of annotated texts that can be used in order to train evaluate on a common basis such learning algorithms on the task.

A major contribution to the development of working SRL systems has been offered by the CoNLL (Computational Natural Language Learning) conference, the annual meeting organized by the Association for Computational Linguistics (ACL) Special Interest Group on Natural Language Learning (SIGNLL).

Since its year 2004 issue, the conference has been running a shared task on semantic role labelling which provided a common ground for SRL systems to be trained and tested against a standardized data set, and to be compared in order to identify the most interesting and promising approaches to the problem.

Given a sentence, the task consists of analysing the propositions expressed by some target verbs of the sentence. In particular, for each target verb all the constituents in the sentence which fill a semantic role of the verb have to be extracted.

The challenge for CoNLL-2004 shared task was to address the SRL problem on the basis of only partial syntactic information, i.e. avoiding the use of full syntactic parse trees and external lexical-semantic knowledge bases. The annotations provided for the development of systems included, the argument boundaries and role labels, words with their POS tags, base chunks, clauses, and named entities [Carreras and Màrquez:2004].

In the 2005 edition, some novelties were introduced [Carreras and Màrquez:2005]:

- the training corpus was substantially enlarged. This allows to test the scalability of learning-based SRL systems to big data sets and compute learning curves to see how much data is necessary to train;
- aiming at evaluating the contribution of full parsing in SRL, the complete syntactic trees given by several alternative parsers was provided as input information for the task;
- in order to test the robustness of the presented systems, a cross-corpora evaluation was performed using fresh test sets from corpora other than the one used for training.

The target dataset of the competition is the PropBank Project [Kingsbury and Palmer:2003], [Kingsbury and Palmer:2002], consisting in a shallow level of predicate argument annotation on top of the syntactic parse trees of the Penn TreeBank [Marcus et al.:1993]. The annotations are inspired by Levin's verb classes of syntactic alternations [Levin:1993]. The dataset is split into 24 sections: sections 01-22 for training, section 24 for development and section 23 for testing.

The participating systems would be trained and tested both on *gold*, i.e. hand-crafted, and automatic syntactic data, to provide respectively an upper bound and a more realistic evaluation of the systems' performance.

Based on the linguistic information provided by a syntactic parser, the modules of a typical software architecture for SRL are responsible for:

- identifying the predicate words that dominate the propositions within a sentence (predicate extraction);
- scanning the parse tree in order to select, for each predicate, the word sequences that are likely to be associated to an argument (candidate argument identification);

- extracting from these data the relevant features for the learning problem (feature extraction);
- classifying these candidate arguments in order to discard those that aren't actual arguments of a predicate (boundary detection). This is generally achieved with a binary classifier that classifies candidates as being arguments of a given predicate or not;
- assigning the proper role label to each argument with a multi-class classifier (argument classification); this can be achieved combining a set of binary margin classifiers (such as Support Vector Machines) either in the Pairwise or the OvA approach [Allwein et al.:2000]. This classification is quite resource consuming, as many binary classifiers have to be trained and employed depending on the number of distinct role labels. Typically, in order to achieve higher efficiency, only those candidate arguments that have passed the boundary classification stage are considered for argument classification;
- performing some joint inference on the resulting labelling schemes in order to resolve conflicts or linguistic inconsistencies. This can be done in many ways, from the application of simple heuristics to the resulting labelling schemes, to the application of techniques inspired by error correction codes, to the employment of complex probabilistic models that consider the likelihood of the whole annotation.

The feature extraction stage is a very delicate one: in order to be properly classified each example has to be represented in the feature space of the learning algorithm, and the quality of this representation has a determinant effect on the resulting accuracy of the learning process. An appropriate description of an example in the feature space presents two major problems:

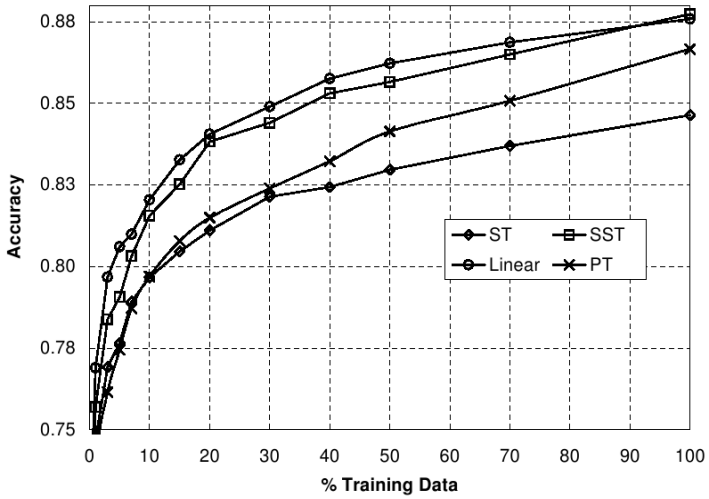
- Given the available data about the examples, i.e. hints on their lexical, syntactic and semantic interpretation eventually available or provided by previous stages of the process, which aspects should be represented in the feature space and which should, instead, be considered irrelevant or poorly representative? (Feature Selection)
- What is the best way to represent these aspects in the target feature space? Which is the most convenient representation with respect to the learning algorithm capability to separate positive and negative examples in the resulting space? (Feature Engineering)

As the recognition of propositions within sentences is a linguistic problem, the features exploited by the learning algorithm describe properties of predicates, arguments and the way they are related. These should provide a linguistic justification of the role of each argument with respect to the predicate in the semantic structure.

There is a large agreement on the effectiveness of a set of linguistic features [Gildea and Jurafsky:2002], [Pradhan et al.:2005] on the local classification tasks of the SRL process, i.e. boundary detection and argument classification, that have been employed in the vast majority of SRL systems. Most of them assume that full syntactic views of the input sentences are available for the SRL task, and associate the arguments of a predicate to those nodes of the parse tree – the argument nodes – that exactly cover the right word sequences. Under this assumption, it is also possible to use tree kernel functions [Collins and Duffy:2002] and directly employ the syntactic parse trees and a selection of their substructures into the learning algorithm.

On the other hand, the selection of linguistic features for more complex, non local tasks such as the joint evaluation of a complete predicate annotation hasn't been thoroughly investigated by the scientific community yet, and there is no such agreement on a set of features to be considered sound and determinant. Especially in this case, a tree kernel based approach is interesting as the learning algorithm can be exploited to automatically select the relevant features which are implicitly encoded by the sentence parse tree.

The University of Tor Vergata took part in the 2005 edition of the CoNLL shared task with a system that employed support vector machines as a learning algorithm for both boundary detection and argument classification, exploiting standard linear features to represent training and test data in the domain of the SVMs [Moschitti et al.:2005] and an OvA combination of SVMs as a role multi classifier for the argument classification stage. The system ranked 8th out of 20 participants from other universities and research centres, with a measured F1 of 75.89% at 3.55% points from the best system [Punyakanok et al.:2005] and 9.16% from the less accurate [Sutton and McCallum:2005].



Since then, we have been improving the accuracy of our SRL system and extending its architecture in many ways, mostly focusing on the employment of tree kernels for the different classification tasks involved. Using tree kernels, we can extract and engineer structural features derived from a syntactic parse tree and directly employ such transformed parse trees for the learning task. The engineering of different structural features for the diverse SRL subtasks constitutes a large part of our research effort and of our contribution to the scientific community researching on SRL. Our hypothesis have always been thoroughly experimented, and the interesting results that we have obtained have been presented in many international machine learning and computational linguistics conferences.

We adopt a two stage structural feature engineering methodology. first, depending on the SRL subtask at study, we project out of the sentence parse tree a substructure of the parse tree which we expect to encode the relevant clues for the learning task. This substructure is eventually transformed in order to render more explicit the belonging of each instance to a class of meaningful structural equivalence which is more strictly correlated with a particular argument, boundary, predicate or labelling scheme. We call this step *canonical mapping*. Second, we modulate the fragment extraction fraction that the tree kernel employs in order to generate the target fragment space. We call this step *feature extraction*.

Canonical mappings aim to capture lexical, syntactic and semantic properties of incoming trees able to feed the learning machine with linguistic effective evidence for the task. One way to achieve this is node marking. A mark-up applied to non terminal nodes NT can generate sub-trees of NT more strictly correlated with a particular linguistic information, i.e. an argument, its boundary or an entire predicate. For example, if we mark the node exactly covering a target argument, the feature extraction will generate substructures from correct and incorrect argument boundaries that are no longer similar. Each mark-up strategy thus results in a kernel function with different number of structures sharable by two trees. The same feature extraction function will thus capture very different types of linguistic similarity. This shows that different kernels can be obtained according to the adopted canonical mappings even with the same algorithms.

For our tests, we employed different configurations of the general system architecture previously described, as required to better stress the aspects concerning the learning problem at study. The empirical evaluations were mostly carried out within the setting defined in the CoNLL-2005 Shared Task. We used as a target dataset the PropBank and the automatic Charniak parse trees of the sentences of Penn TreeBank corpus from the CoNLL 2005 Shared Task data. We employed the SVM-light-TK software which encodes fast tree kernel evaluation [Moschitti et al.:2006a] in the SVM-light software [Joachims:1999]. We used a regularization parameter (option -c) equal to 1 and λ=0.4.

Concerning the employment of tree kernels for SRL, we first run an experiment to select the most promising feature extraction function among those commonly described in literature, i.e. the SubTree (ST), SubSet Tree (SST) and PartialTree (PT) fragment extraction functions [Basili and Moschitti:2005]. The results of this study show that the SST kernel is the more accurate and that the richest kernel in terms of substructures, i.e. the one based on PTs, shows lower accuracy than the SST kernel but higher than the ST kernel.

We then evaluated the impact of tree kernels on the labelling accuracy of an SRL system. We compared on the whole SRL task the F_1 measure of a standard polynomial kernel based on linear features (STD), a tree kernel alone employing a very simple class of structural features called AST_1 and a combination of the two (STD+AST1). The AST_1 is defined as the smallest subtree of the sentences that encompasses an argument node (i.e. a tree node that dominates all only the words of an argument) and the predicate word. The results of this comparison are shown in table:

STD	AST_1	STD+AST_1
75.89	71.00	76.65

We note that:

- standard linear features outperform tree kernels by almost 5 points in F1 , i. e. 75.89 vs. 71. This is due to the fact that in the employed structural features we did not encode very important features like passive voice or predicate position. In [Moschitti:2004], these are shown to be very effective especially when used with a polynomial kernel. Advanced tree kernel engineering may include such and other standard features with a canonical mapping. We have not carried out such study as our aim was to provide new interesting features rather than applying to the simple exercise of representing already designed features within tree kernel functions;
- the combination of the AST1 kernel and the linear features improves the accuracy on the SRL task by almost 0.8 F1 measure points, i.e. 76.65 vs. 75.89. This result suggests that the richer information encoded by AST1 structures can support SVMs in making a correct decision in those cases when linear features are not expressive enough. At the same time, they do not introduce noisy information that would result in an accuracy loss with respect to the STD kernel. Additionally, the result shows that we can exploit previous work in manual feature design in a very easy way.

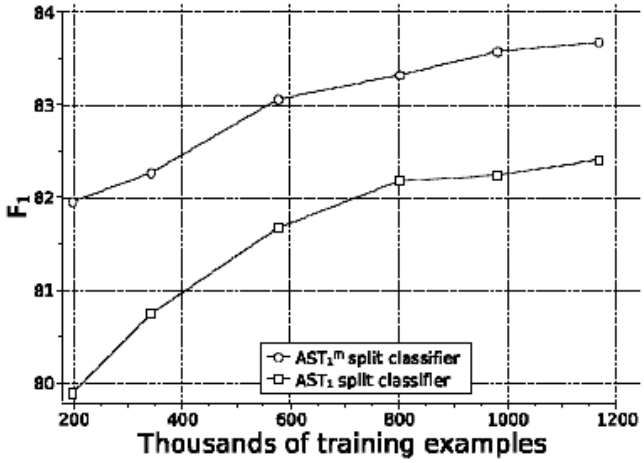
Many experiments have also been run in order to compare the results of the employment of different canonical mappings to the boundary detection and argument classification tasks [Moschitti et al:2006b], which are at the core of our SRL model. We evaluated different strategies for highlighting the target argument node to the learning algorithm, in order to reduce the number of substructures that the tree kernel would match between argument and non-argument nodes, or between argument nodes to be associated with different role labels. Compared to the standard AST_1 structure, our best marking strategy (resulting in a class of structures that we call AST_1^m) shows good results (in terms of F_1 measure) both for the boundary detection and the argument classification tasks:

	Boundary Detection	Argument Classification
AST_1	75.24	75.06
AST_1^m	82.07	77.17

We note that:

a. for boundary detection, AST1m s improve the F1 over AST1 by about 7 F1 points, i. e. 82.07 vs. 75.24: this suggests that marking the argument node simplifies the generalization process;
b. using an engineered tree kernel also improves the argument classification task by about 2 points, i.e. 77.17 vs. 75.06: this confirms the outcome on boundary detection experiments and the fact that we need to distinguish the target node from the others.

We have also derived a learning curve of these two boundary detection kernels, showing that the AST_1^m kernel has a constant accuracy advantage over the AST_1 and that an effective, automatic feature selection can be triggered with training corpora of different sizes:



Boundary detection and argument classification are local learning problems, in that they only classify the individual nodes with respect to the predicate and their position in the sentence, whereas they do not consider the whole predicate argument structure, which is still unknown at node-classification time.

Still, this information is crucial for the accuracy of the resulting labelling for a series of reasons:

- the underlying linguistic model has some constraints that can only be enforced with knowledge about the whole predicate argument structure, e.g. overlapping or nesting arguments are not allowed. As when we detect a boundary we don't about the classification of the other candidates, there must be a way to resolve those cases in which conflicting nodes are both required to be an argument;
- the correct argument structure for a predicate depends on the predicate sentence that is activated in the sentence; the different labels that might be assigned to any candidate argument should be considered in order to guess the current predicate sense and outline the expected argument structure.

For these reasons, many state-of-the-art models perform some joint inference on the whole proposition that is used to refine the output of the local classifiers, which define a possibly raw and inconsistent labelling scheme for the proposition.

We employ a probabilistic inference mechanism which is similar to that described in [Haghighi et al.:2005]:

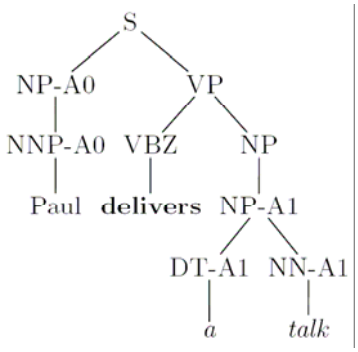
a. we consider all the candidate arguments both for the boundary detection and argument classification subtasks, and store the output of the boundary and role classifiers, for each candidate argument and for each possible role;
b. we provide a probabilistic interpretation of the output of each SVM by mapping the classification scores onto sigmoid distributions [Platt:1999];
c. for each candidate argument, we keep track of the N most likely role labels, including the NARG – not an argument – label;
d. we use a Viterbi algorithm to efficiently explore the space of the possible labelling schemes of each predicate, and select the M most likely configuration. The likely of the annotation is calculated as the product of the probabilities associated to the selected label of each node of the parse tree;
e. we employ an SVM based reranking mechanism in order to compare these M labelling schemes and select the most accurate annotation.

The reranker compares couples <i,j> of possible labelling schemes, each one being described both in terms of linear and structural features. An especially designed tree kernel function combines these representations and, for each input couple, decides whether the left or the right member is more accurate [Moschitti et al.:2006a], [Moschitti et al.:2006b].

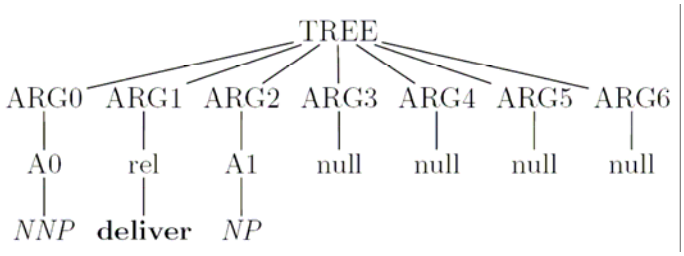
Designing features for proposition reranking is a very difficult task. In fact:

- there is no clear and sound theory describing the relevant features;
- as the sense of the verb is still unknown, it is not possible to be deterministic about the expected argument structure;
- the alternative annotations of a same proposition tend to be quite alike, as the differences generally consist in a changed role label or in very slight movements of the same arguments;
- there is a lot of data to be taken into account, as the whole predicate argument structure generally encompasses a large portion of the encoding parse tree.

Engineered structural features can stress selected aspects of the learning problem, triggering an effective automatic feature selection and improving the accuracy of the reranking mechanism. The structures that we have been employing for the reranking task are very diverse in nature. Some of them are very rich in information, as they preserve most of the syntactic and lexical data encoded by the sentence parse trees. This is the case of the AST_n^{cm} structure, of which an example is shown in figure:



On the other hand, other structures try to encode very little lexical information and to focus on the syntax of the proposition rather than that of the encoding sentence. This is the case of the PAS^{tl} structural feature, of which an example is shown in figure:



The outcome of our reranking experiments have been very encouraging: training the reranker on only 1/20 of the available training material we obtained an F_1 measure on the SRL task of 76.47% using the AST_n^{cm} kernel and of 78.15 using the PAS^{tl} kernel. Training the PAS^{tl} kernel on 1/10 of the available material, the measured accuracy is 78.44%, which is a result in line with state-of-the-art systems that do not employ tree kernels. This last results confirms the difficulty of the proposition reranking task from a machine learning point of view, as adding or removing thousands of training examples has only a small impact on the classification accuracy.

The design of automatic systems for the labelling of semantic roles requires the solution of complex problems. Among others, feature engineering is made difficult by the structural nature of the data, i. e. features should represent information expressed by automatically generated parse trees. This raises two main problems:

1. the modelling of effective features, for some subtasks partially solved in the literature work
2. the implementation of the software for the extraction of a large number of such features.

A system completely (or largely) based on tree kernels alleviate both problems:

1. kernel functions automatically generate features

2. only a procedure for the extraction of subtrees is needed. Although some of the manually designed features seem to be superior to those derived with tree kernels, their combination seems still worth applying.

In these perspectives, we carried out a comprehensive study on the use of tree kernels for semantic role labelling and we designed several canonical mappings associated with the application of innovative tree kernel engineering techniques tailored on different stages of an SRL process.

We tested all the above methods with Support Vector Machines on the data set provided by CoNLL 2005-shared task. The results show that tree kernels suitably engineered always boost the accuracy of the basic systems, as they can either replace or support standard linear features in many SRL subtasks. Most importantly, in complex tasks such as the re-ranking of semantic role annotations they provide an easy way to engineer new features that would be difficult to describe explicitly.

We have shown that tree kernels are a valid support to standard linear features for many stages of the SRL process and engineered different structured features for the classification problems related to SRL, from boundary detection and argument classification, to the re-ranking of complete labelling schemes. Concerning this last point, although we used just a small fraction of the available training data (i. e. 2 sections out of 22 for our latest experiment) the results are very good and encouraging, as our system's accuracy is in line with state of the art systems that do not employ tree kernels.

Although the study that we carried out is quite comprehensive, several issues should be further investigated in the future:

- the tree feature extraction functions ST, SST and PT, should be studied in combination with the proposed canonical mappings. Indeed, the marking strategies may provide more general and effective kernels when applied on the PT space. Moreover, as the PT kernel seems more suitable for the processing of dependency information, it would be interesting to apply it in an architecture using such kind of syntactic parse trees, e.g. [Chen and Rambow:2003]. In particular, the combination of different extraction functions on different syntactic views may lead to very good results;
- once the final set of the most promising kernels is established, we would like to use all the available CoNLL 2005 data. This would allow us to estimate the potentiality of our approach by comparing with literature work on a more fair basis;
- the employment if tree kernels with SVMs on large datasets is very time demanding and results prohibitive on standard hardware. Exploiting tree kernel derived features in a more efficient way, e.g. by selecting the most relevant fragments and using them in an explicit space, is thus an interesting line of future research;

Finally, as CoNLL 2005 [Punyakanok et al.:2005] has shown that multiple parse trees provide the most important boost to the accuracy of SRL systems, we would like extend our model to work with multiple syntactic views of each input sentence.

8 Conclusions

This deliverable discussed the analysis of tools and techniques for cross-linguistic information extraction and retrieval as they were earlier introduced in deliverable 16.3. Aspects related to their support to metadata extraction from heterogeneous material, written in more than one language and including different styles (e.g. ASR transcriptions and Web texts).

The survey of the IE technology adopted in the MAD platform has been focusing on modelling details able to support the system evaluation. The benefits of the extraction tools for metadata discovery for the MAD subsystem has been showed according to a number of experiments and trials described in Section 6. In particular, evaluation experiments for Text categorization, Named Entity Recognition and Classification (NERC) from Web texts as well from ASR transcriptions, Hyperlinking and finally CLIR are reported. The potentials of the technology have been largely analysed and satisfactory outcomes have been derived. It is to be noticed that many evaluation studies reported in this document have been carried out over specific data sets built within the project and their actually limited coverage cannot guarantee the full generality of the reported measures. However, the variety of aspects of the semantic analysis and retrieval of the MAD system have been covered experimentally thanks to the currently complete documentation architecture. The good accuracy reached for most of the subtasks enabled by the current release demonstrate the applicability of the suggested technology in the complex scenario of multimedia annotation and indexing required by Prestospace. Some important extensions possible of the current functionalities are suggested and discussed in Section 7. In particular, more investigation is needed on the performance reachable by the current IR subsystem, that at the actual stage of integration did not allow for extensive evaluation. This will be part of future benchmarking.

9 References

[ACE:2001] http://www.nist.gov/speech/tests/ace/phase2/index.htm

[ACE:2005] http://www.nist.gov/speech/tests/ace/ace05/index.htm

[Agirre and Rigau:1996] Eneko Agirre and German Rigau. 1996. Word sense disambiguation using conceptual density. In Proceedings of COLING'96, pages 16--22, Copenhagen, Danmark

[Allwein et al.:2000] Erin L. Allwein, Robert E. Schapire, Yoram Singer: Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research 1: 113-141 (2000)

[Apt´e et al.:1994] Chidanand Apt´e, Fred Damerau, and Sholom Weiss. Automated learning of decision rules for text categorization. ACM Transactions on Information Systems,12(3):233–251,1994.

[Appelt et al.:1993] D. Appelt and J. Hobbs and J. Bear and D. Israel and M. Kameyama and A. Kehler and D. Martin and K. Meyers and M. Tyson, "SRI International FASTUS system: MUC-6 test results and analysis", In Proceedings of 16th MUC, Columbia, MD. 1993.

[Basili et al.:1998] Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Efficient parsing for information extraction. In Proc. of the ECAI98, Brighton, UK, 1998.

[Basili and Moschitti:2005] Basili, R., A. Moschitti Automatic Text Categorization: From Information Retrieval to Support Vector Learning , Aracne Editrice, Informatica, ISBN: 88-548-0292-1, 2005

[Basili et al.:2004] R. Basili, M. Cammisa, Unsupervised Semantic Disambiguation, Workshop on "Beyond Named Entity Recognition -Semantic Labelling for Natural Language Processing Tasks", held jointly with LREC 2004 LISBON, Portugal, May 2004.

[Bikel et al.:1999] Daniel Bikel, Richard Schwartz, & Ralph M. Weischedel, An Algorithm that Learns What's in a Name" Journal of Machine Learning, vol. 34, n. 1-3, 211-231, (1999).

[Buitelaar:2001] Paul Buitelaar The SENSEVAL-II Panel on Domains, Topics and Senses In: Proceedings of SENSEVAL-II , Toulouse, August, 2001.

[Carreras and Màrquez:2004] Xavier Carreras and Lluís Màrquez, Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In CoNLL-2004, Boston, MA USA. May 2004.

[Carreras and Màrquez:2005] Xavier Carreras and Lluís Màrquez, Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In CoNLL-2005, Ann Arbor, MI USA. June 2005.

[Chen and Rambow:2003] John Chen and Owen Rambow, "Use of Deep Linguistic Features for the Recognition and Labeling of Semantic Arguments", Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003, Sapporo, Japan.

[Charniak:2000] Eugene Charniak. A maximum-entropy-inspired parser. In Proceedings of the 1st Meeting of the North American Chapter of the ACL, pages 132-139, 2000.

[Cohen and Singer:1999] William W. Cohen and Yoram Singer, Context-sensitive learning methods for text categorization, ACM Transactions on Information Systems, 17(2):141–173, 1999.

[Collins:1997] Michael Collins. Three generative, lexicalized models for statistical parsing. In Proceedings of the ACL and EACLinguistics, pages 16-23, Somerset, New Jersey, 1997

[Collins and Duffy:2002] Collins, Michael and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In ACL02.

[Dagan:2000] Ido Dagan, Contextual Word Similarity, in Rob Dale, Hermann Moisl and Harold Somers (Eds.), Handbook of Natural Language Processing, Marcel Dekker Inc, 2000, Chapter 19, pp. 459-476.

[Drucker et al.:1999] Harris Drucker, Support vector machines for spam categorization. IEEE Trans. Neural Netw. 10: 1048–1054.

[Gaizauskas and Wilks:1998] Gaizauskas R. and Wilks Y., Information Extraction: Beyond Document Retrieval, Journal of Documentation, volume 54, 70--105, 1998.

[Gildea and Jurafsky:2002] Gildea, Daniel and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. Computational Linguistics, 28(3):245-288.

[Gliozzo:2005] Alfio Massimiliano Gliozzo, Ph.D. Thesis: "Semantic Domains in Computational Linguistics", Advisor Dr. Carlo Strapparava, University of Trento, December 2005.

[Haghighi et al.:2005] Haghighi, Aria, Kristina Toutanova, and Christopher Manning. 2005. A joint model for semantic role labeling. In Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL2005), pages 173-176, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Ittner et al.:1995] David J. Ittner, David D. Lewis and David D. Ahn, Text categorization of low quality images, In Proceedings of SDAIR-95, pages 301–315, Las Vegas, US, 1995.

[Jelinek: 1999] Fred Jelinek, Statistical Methods for Speech Recognition, MIT Press, 1999.

[Joachims:1998] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of ECML-98, pages 137–142, 1998.

[Joachims:1999] Thorsten Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning, 1999.

[Kingsbury and Palmer:2002] Paul Kingsbury, Martha Palmer, From Treebank to Propbank, Third International Conference on Language Resources and Evaluation, LREC-02, Las Palmas, Canary Islands, Spain, May 28- June 3, 2002.

[Kingsbury and Palmer:2003] Paul Kingsbury and Martha Palmer. PropBank: the Next Level of TreeBank, Proceedings of Treebanks and Lexical Theories '03, Växjö Sweden, 2003.

[Lam and Ho:1998] Lam W and Ho CY, Using a generalized instance set for automatic text categorization. In: Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 81–89, 1998.

[Lewis and Gale:1994] D. Lewis and W. Gale. 1994. A sequential algorithm for training text classifiers. In Proceedings of ACMSIGIR 1994, pages 3-12. ACM-SIGIR.

[Levin:1993] Levin, Beth. 1993. English Verb Classes and Alternations. The University of Chicago Press.

[Lesk:1986] Lesk, Michael E., "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from a Ice-cream Cone," In Proceedings of the ACM SIGDOC Conference, Toronto, Ontario, pp 24-26.

[MUC-3:1991] Proceedings of the 3rd conference on Message understanding 1991, San Diego, California May 21 - 23, 1991.

[MUC-4:1992] Proceedings of the 4th conference on Message understanding 1992, McLean, Virginia June 16 - 18, 1992.

[MUC-5:1993] Proceedings of the 5th conference on Message understanding 1993, Baltimore, Maryland, August 25 - 27, 1993.

[Marcus et al.:1993] Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The Penn Treebank. Computational Linguistics, 19:313-330.

[McCarthy et al.:2004] McCarthy, D., Koeling, R., Weeds, J. and Carroll, J. (2004) Finding predominant senses in untagged text. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. Barcelona, Spain. pp 280-287,ACL Best Paper Award.

[Miller:1990] G. Miller, An On-Line Lexical Database, International Journal of Lexicography, 13,4,235-312, 1990

[Mihalcea:1999] Rada Mihalcea, Word Sense Disambiguation and its Application to Internet Search, Master's Thesis, April 13, 1999.

[Moschitti:2003] Alessandro Moschitti, A study on optimal parameter tuning for Rocchio Text Classifier, in proceedings of the 25th European Conference on Information Retrieval Research (ECIR), Pisa, Italy, 2003.

[Moschitti:2004] Alessandro Moschitti, A study on convolution kernels for shallow semantic parsing. In proceedings of the 42th Conference on Association for Computational Linguistic (ACL2004), Barcelona, Spain.

[Moschitti et al.:2005] Moschitti, Alessandro, Bonaventura Coppola, Daniele Pighin, and Roberto Basili. 2005a. Engineering of syntactic features for shallow semantic parsing. In Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing, pages 48-56, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Moschitti et al.:2006a] Moschitti, Alessandro, Daniele Pighin, and Roberto Basili. 2006. Tree kernel engineering in semantic role labeling systems. In Proceedings of the Workshop on Learning Structured Information in Natural Language Applications, EACL 2006, pages 49-56, Trento, Italy, April. European Chapter of the Association for Computational Linguistics.

[Moschitti et al.:2006b] Alessandro Moschitti, Daniele Pighin and Roberto Basili. Semantic Role Labeling via Tree Kernel joint inference. In Proceedings of the 10th Conference on Computational Natural Language Learning, New York, USA, 2006.

[Morgan:2004] Eric Lease Morgan, Introduction to Search/Retrieve URL Service (SRU), 2004.

[Ng:1996] Ng, Hwee Tou, & Lee, Hian Beng. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (pp. 40-47). Santa Cruz, CA, USA.

[Ng et al.:1997] Hwee Tou Ng, Wei Boon Goh, Kok Leong Low, Feature selection, perception learning, and a usability case study for text categorization, SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, 67--73, ACM Press, 1997.

[Pazienza:1997] Maria Teresa Pazienza, Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97, Frascati, Italy, 14-18, 1997, Springer, Lecture Notes in Computer Science, 1299,ISBN:3-540-63438-X, 1997.

[Pianta et el.:2002] Pianta E, Bentivogli L, Girardi C 2002 MultiWordNet: developing an aligned multilingual database. In Proceedings of the First Global WordNet Conference, Mysore, India.

[Platt:1999] J.C. Platt, Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans, eds., pp. 61-74, MIT Press, 1999.

[Pradhan et al.:2005] Pradhan, Sameer, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support vector learning for semantic argument classification. Machine Learning, 60:13:11-39.

[Punyakanok et al.:2005] Punyakanok, Vasin, Peter Koomen, Dan Roth, and Wentau Yih. 2005. Generalized inference with multiple semantic role labeling systems. In Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL2005), pages 181-184, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Quinlan:1986] J. R. Quinlan, Induction of decision trees. Machine Learning, 1(1), pp 81-106, 2006.

[Sanderson et al.:2005] Sanderson, Robert, with Jeffrey Young and Ralph LeVan, "SRW/U with OAI: Expected and Unexpected Synergies", DLib February 2005

[Schutze and Pedersen:1995] Hinrich Schütze, Jan O. Pedersen, Information retrieval based on word senses, In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, pages 161--175, 1995.

[Schapire et al.:1998] Robert Schapire, Yoram Singer, Amit Singhal, Boosting and Rocchio applied to text filtering, In W. Bruce Croft and Alistair Moffat and Cornelis J. van Rijsbergen and Ross Wilkinson and Justin Zobel, editors, Proceedings of SIGIR-98, pages 215–223, Melbourne, AU, 1998. ACM Press, New York, US.

[Sebastiani:2002] Fabrizio Sebastiani, Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47, 2002.

[Strzalkowski and Jones:1996] Tomek Strzalkowski and Sparck Jones. NLP track at trec-5. In Text REtrieval Conference, 1996.

[Sutton and McCallum:2005] Charles Sutton and Andrew McCallum, Joint Parsing and Semantic Role Labeling, Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), 2005, 225--228, Ann Arbor, Michigan, June, 2005.

[Tzeras and Artman:1993] Konstadinos Tzeras and Stephan Hartmann, Automatic indexing based on Bayesian inference networks, Proceedings of {SIGIR}-93, 16th {ACM} International Conference on Research and Development in Information Retrieval, ACM Press, 22-34, 1993.

[Yarowsky:1992] David Yarowsky. Word sense disambiguation using statistical models of roget's categories trained on large corpora. In Proceedings of COLING '92, Pages 454--460, Nantes, 1992.

[Yarowsky:1995] Yarowsky David, "Unsupervised Word Sense Disambiguation Revealing Supervised methods." In Proceedings of Annual Meeting of the Association of Computational Linguistics, pp189-196.

[Yang:1999] Y. Yang. An evaluation of statistical approaches to text categorization. Information Retrieval Journal, 1999.

[Yang and Chute:1994] Yiming Yang, Christopher G. Chute, An Example-Based Mapping Method for Text Categorization and Retrieval, ACM Transactions on Information Systems, Vol. 12, No. 3, 1994.

Appendix: Editorial Segmentation through Hidden Markov Models

This page is voluntarily left blank.

Modelling the problem of Segmentation of Audiovisual Data as a Machine Learning task

R. Basili

University of Rome Tor Vergata,
Department of Computer Science, Systems and Production,
00133 Roma (Italy),
basili@info.uniroma2.it

Abstract. This short note describes the problem of detecting the correct segmentation of audiovisual data (i.e. TV news formats) and suggests a model to map it into a stochastic process and solve it via a markovian approach. Finally, possible extensions, e.g. the possibilities of applying reranking techniques is discussed. This note is a simple DRAFT and in its current state it is useful to trigger a discussion among the MAD group about the segmentation issue. Mistakes and mispellings are well possible as its completeness has not been addressed yet: plase do not quote it.

1 Introduction

1.1 Problem setting

Given an audiovisual object (AV) d we can define the following set of elements or quantities:

- $T_0^{(d)}$ and $T_\Delta^{(d)} \in \mathbf{R}$ determines the temporal range of d. The (time) length of the range is Δ, i.e. $T_\Delta^{(d)} = T_0^{(d)} + \Delta$.
- P_k is a set of multimedia processors, i.e. content extraction tools[1] that are applicable to the AV data object d. According to the detected content, processors proposes segments as significant information units in the object d: units can be entire news in TV braodcasted news programs or video clips in music programs. Different processors are supposed to detect different sequences of segments according to the heterogeneous information they are tailored to extract (e.g. audio or speech vs. video cues);
- Let $< t_{j_1}^{(k)}, ..., t_{j_n^k}^{(k)} >$ be a sequence of timestamps such that $\forall k, t \quad t_{j_t}^{(k)} \in T_0^{(d)}, T_\Delta^{(d)}$ and $\forall k, t \quad t_{j_t}^{(k)} < t_{j_{t+1}}^{(k)}$. For any processor P_k the sequence $< t_{j_1}^{(k)}, ..., t_{j_n^k}^{(k)} >$ will denote the timestamps at which P_k proposes the end (or beginning) of the current j-th (or next $j+1$-th) segment

[1] GAMPs in the MAD Prestospace platform are typical examples of processors.

Notice how the overall set of timestamps $t_j^{(k)} \quad \forall k, j$ constitute a totally ordered set. Let TS be the set of all the *time ticks* proposed by at least one available processor P_k, i.e.

$$TS = \cup_k \cup_j t_j^{(k)} \tag{1}$$

they constitute a totally ordered set i.e.

$$t_1, ..., t_N,$$

with

$$N = argmax_{k,j}(t_j^{(k)}),$$

and

$$t_i < t_{i+1} \qquad \forall i = 1, ..., N-1$$

Herafter TS will denote the totally ordered sequence $< t_1, ..., t_N >$ over the set of all time ticks $\cup_k \cup_j t_j^{(k)}$.

DEFINTION (*Problem definition*) The problem of finding a correct segmentation for the AV object d can be thus defined as the task of finding a proper subsequence CTS (*Correct Time Stamps*) of TS that optimizes some criteria Γ, i.e.

$$CTS = argmax_{\boldsymbol{s}} \Gamma(\boldsymbol{s}) \tag{2}$$

with $\boldsymbol{s}$ representing a generic subsequence of TS, and Γ a real valued function that expresses the quality of a subsequence. In principle the segmentation problem so defined reduces to the design of a proper definition for the function Γ.

1.2 Properties of a timestamp sequence

Every processor P_k is generating the sequence $< t_{j_1}^{(k)}, ..., t_{j_n^k}^{(k)} >$ according to some source information:

- a specific media level (e.g. audio channels for the ASR) that also characterizes the data format of the processor input (e.g. source sound of the spoken information)
- a specific set of features observable for the input data of P_k (e.g. energy or volume of the sound)
- a specific set of criteria for generating the timestamps, i.e. the decision to end/initiate a segment at $t_{j_n}^{(k)}$ (e.g. the threshold of the volume under which silence is assumed and a break, i.e. a change of focus, in the spoken text motivates a segment's ending).

The decision criteria of processors are thus functions of properties and depends on the short intervals of time before and after individual time ticks $t_{j_n}^{(k)}$. We will denote these individual intervals as $\Delta(t_j^{(k)})$, or more simply Δ_j when the dependency from the individual processor P_k can be neglected.

Although the problem of segmentation should be clearly separated from the individual work of the processors, it is obvious that a good Γ depends from the individual features and criteria adopted by the processors P_k. In order to detach the modelling of the segmentation problem from the individual work of processors P_k, we define the notion of generic properties for a processor P_k during the time interval $\Delta(t_j^{(k)})$ related to the j-th segmentation time point. This generic notion of property in the short interval Δ_j will be used and it will be denoted $\pi(\Delta_j^k)$, or more simply π_j^k.

2 Machine learning for audiovisual segmentation

The modeling of the function Γ can be obtained in two major ways:

- A model that depends on the general study of the processors on a restricted data sets and defines (at design time) empirically all the paremeters of the Γ function. We will refer to this heuristic model as the *design-time model*.
- A model that depends on the behaviour of the individual processors on real data sets as those that can be observed in the normal functioning of the target AV system. Observation are assumed to be correct (without mistakes, i.e. manually corrected): in this scenario, the segmentation time points $t_1, ..., t_N$ of the processed AV documents d are all manually labelled as *correct* or *incorrect*. The Γ function can then be derived via general induction algorithms (e.g. bayesian probabilities or Markov processes) over the labeled segments data sets. We will refer to this model as the *adaptive model*.

2.1 The heuristic *design-time* model

The *design-time model* can be defined as a simple combination of the output of individual processors P_k. Let $P_k(t_i)$ the boolean function describing the individual decisions of processors P_k on time point t_i: $P_k(t_i)$ will be 1 if the k-th processor proposes a segment end in t_i (or, equivalently, a segment start) and 0 otherwise.

An approach via a simple voting can be applied. In this case the function $\Gamma(t_1, ..., t_N)$ is realized by a set (sequence) of N boolean decision functions $\Gamma_i(t_i)$ acting on individual time points, t_i: the final segmentation is obtained as the subsequence $< t_{i_1}, ..., t_{i_n} >$ of the original candidate $< t_1, ..., t_N >$ such that:

$$\begin{array}{lll} \forall m = 1, ..., n < N & \exists j \in \{1, ..., N\} & \text{with } t_{i_m} = t_j \\ & \forall m = 1, ..., n & \Gamma_{i_m}(t_{i_m}) = true \end{array} \quad (3)$$

and

$$\forall t_j \notin < t_{i_1}, ..., t_{i_n} > \qquad \Gamma_j(t_j) = false$$

DEFINITION (*Simple Voting*). A simple definition for Γ_i can be obtained as the voting among the K individual processors, i.e.

$$\Gamma_{i_m}(t_{i_m}) = \begin{cases} true & \text{if } \sum_k P_k(t_{i_m}) > \alpha \cdot K \\ false & otherwise \end{cases} \tag{4}$$

where $\sum_k P_k(t_{i_m})$ defines the number of processors P_k that accept the time point t_{i_m} as a break between two segments, and $\alpha \in [0, 1]$ is the threshold for acceptance, i.e. the percentage among the pool of K processors. Equation 4 defines the *simple voting scheme* for the individual $\Gamma_i(t)$.

A variant of the Equation 4 can be obtained by weighting the ouput of individual processors P_k according to a given criteria, σ. This allows to assign more importance to some processors and reduces the impact of other ones on the final decision (as defined in Eq. 4). Higher weights, for example, should be given to more performant processors, via individual σ_k.

DEFINITION (*Weighted voting*). Given an array of K parameters $\sigma_1, ..., \sigma_K$ (weights for the individual processors) such that $\sum_k \sigma_k = 1$, a *weighted voting policy* is defined as follows:

$$\Gamma_{i_m}(t_{i_m}) = \begin{cases} true & \text{if } \sum_k \sigma_k \cdot P_k(t_{i_m}) > \alpha \cdot K \\ false & otherwise \end{cases} \tag{5}$$

Equation 5 defines a general weighted voting algorithm that determines the target subsequence CTS: time ticks t_{i_m} are in CTS, only if $\Gamma_{i_m}(t_{i_m}) = true$.

The parameters σ_k of this model are weights (e.g. reliability factors) assigned to the individual categorizers P_k. Although weights σ_k can be assigned on several basis, a simple estimate is to associate them the average performance of P_k. This can be done on a controlled data set or simply as an empirical estimate. In this case the higher is the average performance of a processor P_k the closer to 1 its weight σ_k will be. Output of more performant processors P_k will be more influential on the decision implemented by the Γ_i fuction of Eq. 5.

Scores σ_k express the general *reliability* of a processor (independently from the different time points t). Reliability in this way is an *external* property of a processor as it depends on its general behavious on (usually large) test data sets. An issue orthogonal to reliability is the internal notion of *confidence*. Confidence captures the strength by which individual processors P_k decide to create a given segment in time point t_i. It is thus a model of the internal state of each P_k in the short time Δ_i around t_i.

We can model the confidence by which a processor decides the start of a given segment at a generic time point t_i as a real valued function over time.

DEFINITION (*Extended Weighted Voting*) Given a set of real-valued functions $P_k(t)$ defining the outcome of individual processors P_k on time points t, and ranging over $[0, 1]$, i.e.

$$\forall k, t \quad P_k(t) \in [0, 1],$$

an extension of the weighted voting scheme is given by:

$$\Gamma_{i_m}(t_{i_m}) = \begin{cases} true & \text{if } \sum_k \sigma_k \cdot P_k(t_{i_m}) > \beta \\ false & otherwise \end{cases} \tag{6}$$

where $\sum_k \sigma_k = 1$ and β is a threshold that has to be empirically determined.

Eq. 6 provides a more expressive model taking into account the general reliability of processors P_k as well as their internal status related to individual decisions, i.e. starts of new segments. The general problem of detecting the proper parameter settings for Eq. 6 (i.e. the estimation of optimal σ_k and β) is however more complex: individual σ_k may be in fact dependent on time t as they are tight to the confidence scores $P_k(t)$ at individual time points t.

In order to compute the quantities in Eq. 4, 5 and 6 every function $P_k(t_i)$ must be defined at every generic time point t_i in $< t_1, ..., t_N >$. While the boolean versions of functions $P_k(t_i)$ are always defined, this does not hold for Eq. 6 modeling the confidence-based voting mechanism. In fact, a generic processor P_k does not output confidence values in every time point t_i: confidence values $P_k(t_i)$ are available only when P_k takes a decision, i.e. suggests the start of a new segment. Let us call the set of the other time points in $< t_1, ..., t_N >$ as the *silence set* S_k of a processor P_k. In the silence set, i.e. for $t_i \in S_k$, the confidence value $P_k(t_i)$ is unknown. A suitable extension $\hat{P}_k(t)$ for $P_k(t)$ in S_k should thus be found.

DEFINITION (*Extended confidence scores*). For every processor P_k,given its silence set S_k and its complement in TS $\overline{S_k}$, the following extensions of the confidence functions

are defined:

$$\forall k, \forall t_i \in S_k \quad \hat{P}_k(t_i) = 0 \quad \text{(Full confidence)} \tag{7}$$

$$\forall k, \forall t_i \in S_k \quad \hat{P}_k(t_i) = \mu^{(k)} \quad \text{(Average confidence)} \tag{8}$$

$$\forall k, \forall t_i \in S_k \; \hat{P}_k(t_i) = 1 - \mu^{(k)} \quad \text{(Complement confidence)} \tag{9}$$

where $\mu^{(k)}$ is the average of the observable confidence scores for processor P_k (in $\overline{S_k}$), i.e.

$$\mu^{(k)} = \frac{\sum_{t_j \in \overline{S_k}} P_k(t_j)}{|\overline{S_k}|} \tag{10}$$

A revised definition of the extended weighted voting scheme can be thus given by the following equation that replaces Eq. 6:

$$\Gamma_{i_m}(t_{i_m}) = \begin{cases} true & \text{if } \sum_k \sigma_k \cdot \hat{P}_k(t_{i_m}) > \beta \\ false & otherwise \end{cases} \tag{11}$$

where, again, $\sum_k \sigma_k = 1$ and β is an empirical threshold.

The Equations 4, 5 and 11 provide different models for a segmentation algorithm based on a pool of media-specific processes P_k. The ability of modeling reliability and confidence allows also to model in a better way the individual properties and define a richer global segmentation scheme. Limitations of the above approaches are related to the following issues:

- The problems of modeling in an effective way the dependencies of *confidence* and *reliability* over time. Functions σ_k, $P_k(t)$ and $\hat{P}_k(t)$ are infact rough estimates: constants, as the weights σ_k expresses a generic notion (e.g. the average) of performance (scores); interpolations of observable properties as guesses $\hat{P}_k(t)$ of confidence values in *hidden* time points as in Eq. 7-9 are required.

- The inability of the system to combine properly the full range of information available from the processors P_k. Every function acts only on the basis of individual decisions (i.e. the outcome $P_k(t)$) due to relative simplicity of the combination (*voting*) scheme. Relative dependencies among different processors are not taken into account by the combination function: it uses the individual output scores ($P_k(t)$) as independent functions. Each $\Gamma_i(t_i)$ function takes a decision only according to the state of a single individual processor P_k in the small interval Δ_i around a time point t_i: no dependency among the processor's decision is exploited in the voting scheme

– The inability of the system to exploit global information. Individal decisions are independent each other, so that decision at time point t_i is not influenced by the decisions taken in the preceeding (or forthcoming) time points $(t_{i-1}, t_{i-2}, ...)$. Although programs have a relatively well known format this information cannot be exploited properly at the level of individual segmentation decision, t_i. For example, there is no way to express the desired average number of segments of the resulting optimal sequence CTS, although it does not change for large classes of TV broadcasted news. Moreover, the largely regular design of sequence of segments (e.g. some topics in TV news are regularly treated before others and the average occurrence of some topics is relatively stable over time) cannot be modeled in the proposed voting schemes.

The third observation suggests that the optimal sequence should not be found as a cascade of local decisions: this is the process implied by Eq. 3 where functions $\Gamma_i(t_i)$ act as masks for the incoming overall sequence TS but work in isolation on individual time points t_i. A better approach is to select the optimal (sub)sequence CTS among candidates depending (at the same time) on the entire $TS =< t_1, ..., t_N >$. Making the chain of decisions related to increasing time points t_i sensible to the entire sequence can be easily obtained via a finite-state probabilstic model, i.e. a Markov chain.

3 Sequences of Temporal segments as stochastic processes

In probabilistic terms the segmentation problem can be seen as the detection of the most likely chain of decisions that take place at all the individual time points, $t_1, ..., t_N$. Each decision at t_i accepts or rejects the proposal of one or more processors P_k. As the existence of a segment end (or start) is uncertain we can model individual decisions as random variables X_i $(i = 1, ..., N)$ taking boolean values in the $\{true, false\}$ set. Under this perspective we can say that the likelyhood of a segment end (or start) at t_i is given by the probability $p(X_i = true)$ (or $1 - p(X_i = false)$. The overall decision (i.e. the computation of segments for the entire sequence) is thus modeled as the maximization of the probabilty for a sequence of random variables

$$< X_i, ..., X_N >:$$

The uncertainty underlying the problem is related to the unknown (optimal) decision at time point t_i justified only by the AV object, i.e. the content of the target program. The only information available about content is provided by the processors: at each time point t_i we have information about

– reliability of individual processors
– confidence
– other processor specific properties (e.g. complexity of individual pieces of information, volume levels, lexical cohesion, ...)

Such information may be denoted as π_{t_i} as a synthetic representation of individual properties $\pi_{t_i}^{(k)}$ of processors P_k: we can juxtapose all $\pi_{t_i}^{(k)}$ to get unique feature vectors π_{t_i}. As an example, we can use just the reliability scores σ_k and confidence values $P_k(t_i)$ discussed in the previous sections: this results in the following definition for π_{t_i}:
$\pi_{t_i} =< \sigma_1, P_1(t_i), \sigma_2, P_2(t_i), ..., \sigma_K, P_K(t_i) >$
π_{t_i} is the only *visible* information about the segmentation problem as the rest (i.e. the program changes of state) is hidden.

In this case the overall problem can be modeled as follows:

DEFINITION (*Probabilistic problem setting*). Finding the optimal segmentation sequence CTS within the original one $TS =< t_1, ..., t_N >$ is equivalent to maximize the likelyhood of the chain of changes of state (i.e. the program progresses from t_i to t_{i+1}) given the visible information guaranteed by the processors. Formally we need to compute the following function:

$$CTS = argmax_{d_1,...,d_N} \quad p(X_1 = d_1, ..., X_N = d_N | \pi_1, ..., \pi_N) \tag{12}$$

or more simply

$$CTS = argmax_{d_1,...,d_N} \quad p(d_1, ..., d_N | \pi_1, ..., \pi_N)$$

where

$$\text{States: } d_i \in \{true, false\} \text{ or, equivalently, } \{0, 1\}$$
$$\text{Visible Output: } \pi_i = \pi_{t_i} \text{ (the output signal)}$$

The above modeling implicitly assumes that:

- The TV program is a source of information that changes state over time. Possible changes are $true$ (i.e. a segment needs to be started) or $false$ (i.e. continue the previous segment)
- Segmentation points are related to changes of state of the emitting source: some changes (i.e. time points) determine segment starts while others do not. This mechanism works as a selection criteria of the optimal sequence CTS within the overall TS: when $X_i = true$ t_i from the input TS should be inserted in the targer CTS
- If the source is in a given state it emits observable messages π_i (i.e. properties of the processor pool): the law by which observable messages are emitted from the source state is hidden. It is thus unknown how likely is a segment start given an observed message π_i.

First of all, notice how Eq. 12 can be equivalently rewritten (by the Bayes law) as:

$$CTS = argmax_{d_1,...,d_N} \quad p(\pi_1, ..., \pi_N | d_1, ..., d_N) p(d_1, ..., d_N) \tag{13}$$

so that the *emission* phenomena and the *transitions* between states are more clearly separated:

- Emission is modeled by probablities of observing properties π_i when the source is in state d_i, i.e. $p(\pi_i|d_i)$
- Transitions are modelled via conditional probabilities of $true|false$ assignments in the changes of states (random variables X_i), i.e. $p(X_i = d_i|X_{i-1} = d_{i-1}, ...)$

The above model maps a segmentation problem into an *hidden markov model*, i.e. a technique for reducing an optimization problem to a finite-state probabilstic process. Markov models are described in a rich scientific literature whose main results make availabile:

- efficient methodologies for solving the optimization problem defined in 13 (*Statistical Inference*)
- robust estimation methods to effectively compute the model parameters (e.g. the transition and the emission probabilities)
- a wide variety of software tools that implement the Markov modeling and its estimation and inferece procedures[2].

We will discuss hereafter the implications of the adoption of HHMs for the segmentation problem. First of all, HHM gives us the possibility of describing in the same model the uncertainty in the state transitions, i.e. in the segmentation decisions, and the uncertainty in the right observation, i.e. the properties π_i of the pool of processors in the short time interval around a time point t_i. Moreover, the dependencies in the chain of decisions are also taken into account. Infact Eq. 13 defines the maximization of the likelyhood for the entire chain $X_1, ..., X_N$ so that global constraints are applied. Some basics about Markov models will also clarify how general techniques (e.g. the Viterbi algorithm for the Statistical Inference) can be applied efficiently to the problem.

3.1 Markov Models

Suppose $X_1, X_2, ..., X_N$ forms a sequence of random variables taking values in a countable set $W = d_1, d_2, ..., d_M$ (State space). If the following properties apply the sequence of $X_1, X_2, ..., X_N$ is a **Markov chain**:

[2] See for example the Java library at `http://www.run.montefiore.ulg.ac.be/ francois/software/jahmm/`

– Limited Horizon Property:
$p(X_{i+1} = d_k|X_1, ..., X_i) = p(X_{i+1} = d_k|X_i)$
In other words, state at time point t_{i+1} depends only on the previous time point t_i.

– Time invariance:
$p(X_{i+1} = d_k|X_i = d_l) = p(X_2 = d_k|X_1 = d_l) \qquad \forall i(> 1)$

The limited horizon property implies that the changes of state depend on a limited memory: in particular, in our definition, *only* from the previous state. These models are also called *memoryless*. Notice that this assumption can be relaxed by modelling in a state more complex information, i.e. more history along the chain. For example we could map the initial problem with random variables $X_1, ..., X_N$ into a new chain $Y1, ..., Y_{N-1}$ where random variables Y_i represent pairs of random variables $< X_{i-1}, X_i >$ so that transitions between Y_i's are more informed about the chain history (as they depend on the two most recent time points).

In order to represent a Markov Chain we need:

– The **transition matrix** A:

$$a_{kj} = p(X_{i+1} = d_j|X_i = d_k)$$

Note that $\forall k, j \quad a_{kj} \geq 0$ and $\forall k \quad \sum_j a_{kj} = 1$. The probability of all the (possible) transitions out from one state k is total.

– The **initial state description** (i.e. probabilities of initial states):

$$p_i^{(1)} = p(X_1 = d_i)$$

Note that $\sum_{j=1}^{M} p_j^{(1)} = 1$, i.e. the State space fits the entire spectrum of possibilities for the status of the emitting source at the beginning. In the segmentation problem, the State space is $\{0, 1\}$ so that $p_1^{(1)}$=1 and $p_0^{(1)}$=0: the starting time of the program always establishes the first segment start.

A Markov chain is defined as an **Hidden** Markov model if the emitted signals (i.e. the visible information) is uncertain with respect to the individual states X_i. What we need is a further random variable O that describes the uncertain observed emissions π_i: the distribution is thus $p(O_i = \pi_k)$.

3.2 The HMM Model of the segmentation task

In this section we are discussing a model that maps the segmentation task into a stochastic process. The main assumptions behind it are:

– **HMM States are mapped into segmentation decisions at time stamps** (d_k), so that

$$p(d_1, ..., d_n) = p^{(1)}(d_1)p(d_2|d_1)...p(d_N|d_{N-1})$$

– **HMM Output symbols are the observable properties** (π_i), so that

$$p(\pi_1, ..., \pi_N|d_1, ..., d_n) \quad = \quad \prod_{i=1}^{N} p(\pi_i|d_i)$$

– **Transitions** represent *moves* from one decision to the next one

Note that *the Markov assumption is used*:

– to model probability of a decision in time point t_i (i.e. d_i) only as a function of the preceeding decisions (i.e. d_{i-1})

– to model probabilities of properties (i.e. π_i) based only on the current state (i.e. decision d_i appearing in time point t_i).

The above assumptions allow to limit the number of parameters of the model as only the transition matrix and the output probabilites are needed.

3.3 Implementation Issues

The application of the Hidden Markok Model defined by Eq. 13 depends on two probelms:

– *Statistical Inference*, i.e. the efficient computation of the optimization problem in the equation
– *Parameter estimation*, i.e. the definition of the transition matrix and the output probabilities. Notice that in the segmentation problem the initial probablities are easily defined as follows: $p_1^{(1)}$=1 and $p_0^{(1)}$=0

The problem of statistical inference is solved by the well-known Viterbi algorithm. The algorithm solve the problem of searching the likeliest State transition sequence[3]. The Viterbi algorithm finds the sequence as a path in the graphical representation of the Markov model: a *trellis*. In this graph, nodes are the possible states in the N different time points while arcs are transitions among states/nodes adjacent (along time). At each step, the best path (and continuations) are considered up to the reached time point. A dynamic programming technique is here applied where buffer variables are precomputed

[3] It makes the computational complexity quadratic in the number of states and time points

(inside and outside probabilties): partial probability values are thus stored at th enodes and made available without overhead. The result is the best path, that is the walk across the graph that maximizes the likelyhood of the entire chain. The wide availablity of software tools for the general solution of a Markov chain solve the first implementation problem of our segmentation task.

Estimation of the required parameters can be afforded in two major ways: supervised methods based on hand-annotated training data and unsupervised methods.

The traditional approach (i.e. the supervised one) foresees the availability of controlled data where individual states/decisions are available along with the emitted signals. For example, HMMs have been adopted for the text processing task known as *Part-of-speech* (POS) tagging. Given a text POS tagging results in the labeling of individual words (tokens) with their major grammatical categories (nouns, vs. verbs, or adjectives). The syntactic ambiguity of words in natural language makes this task very complex when applied to general texts. The application of HHM maps states into syntactic labels and words into the emitted (observable) signals. In this case, the parameter estimation is directly enabled by hand labelled data sets. Text collections are here examined and all individual tokens are given the correct label by linguists. In this way transition probabilities (i.e. $p(d_2|d_1)$) are obtained by counting how many times some labels (i.e. d_2) follows immediately after the others (i.e. d_1) in the annotated texts. Output probablilities (i.e. $p(\pi_k|d_k)$) are obtained by counting how many times a given syntactic label (d_k) is found in the corpus attached to the words, π_k. All these probabilities can be easily obtained from annotated corpora and the estimation task is rather simple. Problems arise from lack of information (e.g. words never encountered in the annotated collection) or low counts (e.g. poor estimates are obtained for very rare words). Both the issues inspired a large literature to tackle this problem, often referred to as *data sparseness*. Rare phenomena characterize small collections where data are too sparse and frequency counts do not result in reliable probability estimates. In this case smoothing techniques are adopted: probabilities of phenomena that are rare or never appearing in the training collection are adjusted in order to provide more robust information. Major adjustment (*smoothing*) techniques are *discounting*, *Good-Turing* estimation or *back-off*. These techniques allows faster training and reach reasonable performance levels even when only small training data sets are available.

Probability estimation for the segmentation problem requires thus the annotation of a consistent set of TV programs with the related segments. The time points involved (i.e. those proposed by at least one processor P_k) are analysed and the right ones are defined (i.e. labeled by *true* during the validation/documentation process). This information is sufficient to collect :

- *transition probabililities* as they corresponds to the number of times adjacent pairs of decision $true|false$ appear in the annotated programs

– emission (or output) probabilities amount to the number of times a given state d_k is found in correspondance with a given set of properties π_i. In this case, the larger is the number of different properties adopted, the stronger is the requirement on the size of the annotated data set (i.e. the number of TV programs that need to be annotated for reliable estimates). However even complex features/properties may be modelled via a smal number of discrete values. For example, few classes (degrees) of confidence values can effectively represent the variations of the $P_k(t)$ values in the $[0, 1]$ interval. Discretizing the properties defined in π_k is a simple way to preserve useful information about segments and processors P_k by also minimizing the training requirements of the Markov model. Furthmore, the adoption of smoothing techniquesis is even more beneficial as it allows better inferences when small training material is available.

A second approach to the parameter estimation is unsupervised and makes use of the algorithm called Expectation Maximization. These algorithms can be fed with random assignment of values to the required probabilties and then proceed incrementally to refine the values in order to better fit existing (non annotated) data. Although they suffer from local minima and overfitting problems (this may limit their impact on the accuracy of predictions), they are able to trigger the segmentation process very soon and support effctive ways of incrementally and semi-automatically build growing portions of hand annotated (i.e. training) data sets.

3.4 An alternative markov model for segmentation

The simple model introduced in the previous section can be further refined in order to better exploit the available information from processors P_k. Currently we foresee two possible states in each time point t_i: $true$ and $false$. This means that the possibilities of modelling state changes are very poor, i.e. only probabilities like $p(X_i = true|X_{i-1} = false)$ are used. One of the missing information from this model is the processor P_k responsible for a given break.

What we would like to exploit is facts like the following:

– when a given processor (e.g. P_j) says $false$ in a time point, it is more likely that it will say $true$ in next time point;
– correlations between processors: for example, the audio information (i.e. processor P_{audio}) may preceede the video one (i.e. processor P_{video}). The break of the news reading at a time point t_i may happen significantly before the camera changes at the video level that happen at time point t_{i+1}. In this case the former information should increase the evidence of the latter, i.e. we expect higher probabilties of $true$ at time stamp t_{i+1} rather than at t_i.

Notice how the presented model cannot precisely express these kinds of information as transition probabilities $p(X_{i+1} = true|X_i = false)$ do not depend on individual processors.

A way to solve the above problem would be to represent in the states of the Markov chain not only the acceptance or rejection of segment's starts (ends) at time point t_i, but also the responsible processors. Remember that the state chains $d_1, ..., d_N$ are the final result of the Markov model as a solution of the statistical inference process in Eq. 13. The boolean values, i.e. $d_j \in \{true, false\}$) allows to select all and only the correct (i.e. more likely according to the model) time points from the initial overall sequence TS. Although more than one processor can suggest the same segment, what we need here is to express which processor P_k expresses which decision at a time stamp t_i. This can be easily done by extending the State space W, i.e. the domain of the d_k values. Instead of using the boolean set $\{true, false\}$ we can define the values d_{kj} in the following way:

$$\forall \text{ processor } k, \quad d_{kj} = \begin{cases} true & \text{iff } j = 1 \\ false & \text{iff } j = 0 \end{cases}$$

In this way in time points t_i, the assignments of random variables $X_i = d_{kj}$ express the information that a processor P_k accepted (i.e. j=1) or rejected (j=0) a segment start in time point t_i. Transitions probabilities like $p(X_{i+1} = d_{k_1,j_1} | X_i = d_{k_2,j_2})$ now can capture dependencies between processor's (i.e. P_{k_1} and P_{k_2}) decisions.