A2Q: an agent-based architecure for multilingual Q&A

Roberto Basili, Nicola Lorusso, Maria Teresa Pazienza, Fabio Massimo Zanzotto

Dipartimento di Informatica, Sistemi e Produzione
Universitá di Roma "Tor Vergata",
00133 Roma
{basili,pazienza,zanzotto}@info.uniroma2.it

Abstract

In this paper we describe the agent based architecture and extensively report the design of the shallow processing model in it. We present the general model describing the data flow and the expected activities that have to be carried out. The notion of *question session* will be introduced as a means to control the communication among the different agents. We then present a shallow model mainly based on an IR engine and a passage re-ranking that uses the notion of expanded query. We will report the pilot investigation on the performances of the method.

1. Introduction

Question-answering (QA) is a field in which a number of different techniques have to work cooperatively in order to achieve the correct final result given the user input query. The architectural design of QA systems is then a crucial point. The requirements that QA systems have to fulfil are very diverse and eventually inconsistent. On the one hand, they have to answer to any factoid question in an open domain: this requires the adoption of very robust Information Retrieval oriented techniques. On the other hand, the extraction of very specific text fragment is also expected implying the use of some complex language model.

Given, for instance, a very specific, i.e. not based on "general" knowledge, factoid question such as *"Which company had a positive net income in the financial year 2001?"* where the expected answer is rare, each bit of relevant text has to be investigated. Therefore, a QA system in this case has to exploit the equivalence between this linguistic form and the one in the following text fragment: *"Acme Inc. reported revenues of $.9 million for the year ended in December 2001."*. Such an equivalence between linguistic realisations of relational concepts are very relevant bits of semantic dictionaries and are often called *semantic frames*. In the example, the relational concept *have-revenues(AGENT:X, AMOUNT:Y, TIME:Z)* can describe a generalised level in which the two related linguistic forms *X has a positive net income of Y in Z* and *X reports revenues of Y for Z* are equivalent.

Generally, semantic-oriented applications such as Information Extraction rely on complete semantic models consisting of:

- a catalogue of named entity classes (relevant concepts) as *Company*, *Currency*, and *TimePeriod*;
- a catalogue of (generally) coarse-grained relational concepts with their semantic restrictions, e.g. *have-revenues(AGENT:Company, AMOUNT:Currency, TIME:TimePeriod)*;
- a set of rules for detecting named entities realised in texts and assigning them to the correct class;
- a catalogue of one-to-many mappings between the coarse-grained relational concepts and the corresponding linguistic realisations.

These semantic models are often organised using logical formalisms (as in (Gaizauskas and Humphreys, 1997)). The results are very interesting artefacts conceived to represent equivalences among linguistic forms in a systematic and principled manner.

In a QA system such a model has to find its place in cooperation with the bag-of-word oriented models that gives the necessary coverage for the overall system. The wide-coverage and the very specific extraction are contrasting requirements for different processing modules. The solution we adopt is a agent architecture (based on JADE (Bellifemine et al., 1999)) in which different processing techniques are spread through a pool of software components that cooperate for the final goal. Each important processing capability is embedded into an agent. Cooperative work is then coordinated by a central agent that plays the role of an "activity scheduler". In order to keep the consistency of the cooperative processing the information is represented by a common data structure, that we call "question session", shared among the agents. The resulting architecture is able to host two different ways of processing the input question:

- a light processing mode, mainly embodied by a search engine and a shallow sentence processing and answer matching technique;
- a knowledge-intensive question processing model based on ontological information over a domain and making use of inference techniques for answer matching and interpretation

In this paper we describe the agent based architecture and extensively report the design of the shallow processing model in it. First of all in Sec. 2. we present the general model describing the data flow and the expected activities that have to be carried out. Each activity will be carried out by a single agent. Here the notion of *question session* will be introduced as a means to control the communication among the diffent agents. In Sec. 3., we present a shallow model mainly based on an IR engine and a passage re-ranking that uses the notion of expanded query. We

will report the pilot investigation on the performances of the method.

2. An agent-based architecture for Q&A

An agent-based architecture seems to be the correct answer for meeting the very diverse and eventually inconsistent requirements. Very different systems may work cooperatively in order to achieve the final goal of answering input questions. However, agent-based architectures may result in places with an high entropy if not correctly governed. Agents in the architecture shuold have a clear role and should communicate with a shared language. What we propose here is a master-slave approach in which each slave agent has a very specific role to play and its activation is governed by the master agent.

This architecture should meet the overall goal to answer input questions using knowledge stored in document collections or in the web. As depicted in Fig. 1, it has to host a number of "classical" subtasks as the construction of the query for the information retrieval engine (*Query Generation*), the re-ranking of the documents given by the IR engine with respect to considerations that are not made on the bag-of-word model (*Document Reranking*), the selection of the relevant passages in the selected documents (*Passage Retrieval*), and the actual extraction of the answer out from the selected passages (*Answer Matching and Extraction*).

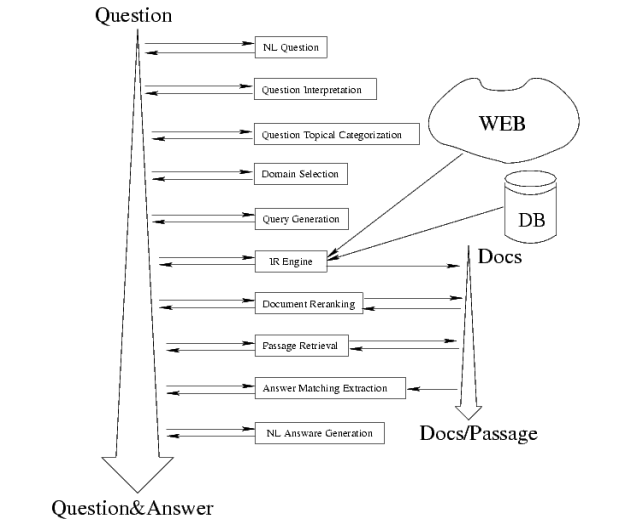

Figure 1: Q&A Data Flow and Activities

Moreover, the notion of knowledge domain may be very useful both in open-domain question answering and when specific ontological domain models are available. Under this perspective, questions (and eventually documents) have to be classified according to a topical classification scheme. This requires a specific activity that should be hosted in the architecture (i.e. *Question Topical Categorisation*).

In an agent based architecture, each of these activities has to be performed by a specific agent and, in order to guarantee the achievement of an answer, each activity may be performed by different agents adopting different technologies. Keeping the consistency in the communication and in the cooperation among these agents is a very difficult objective.

However, as depicted in Fig. 1, the information flow among the different activities embodies by agents is clear and may be interpreted along two different main axes: the state of the question processing activity (the arrow *Question* to *Question&Answer*) and the state of the source of the information (the arrow *Docs* to *Docs/Passages*). The first flow describes the changing of the information gathered around the question while the second takes into account the documents suspected to contain the answer to the input question.

The consistency among the different agents may be kept if a clear language for these two information flows is defined. The model of the language, called *question session*, is defined in Sec. 2.1..

It is worth noticing that an agent based architecture is naturally open to the distribution of the workload on a network of computational resources. The JADE agent platform (Bellifemine et al., 1999) easily supports this facility as agents may be allocated in any machine in a specific network.

2.1. A unifying data model: The Question Session

The agent communication language is realised on a unifying data model that keeps track of the status of the analysis of the input question. This data model, called *question session*, gives the possibility of implementing Q&A systems ranging between the two extremes: knowledge-based approaches and IR-based approaches.

The question session contains therefore the following information:

- *Question*: the input question as it is
- *Question-XDG*: a syntactic representation of the input question expressed in the eXtended Dependency Graph (XDG) formalism (Basili and Zanzotto, 2002)
- *Question-QLF*: a syntactic/quasi-logical representation of the input question (Gaizauskas and Humphreys, 1997) that should enable the semantic analysis whenever necessary
- *Focuses*: one or more focuses of the input question intended as the phrases disambiguating the question and able to emphasise the answer
- *Categories*: one or more topical categories of the input question that may be used to select relevant documents where the answer can be found or relevant domain knowledge bases
- *Query*: the query related to the input question that has to be posed to the underlying information retrieval engine
- *Documents*: the list of documents ordered according to the relative relevance with respect to the input question
- *Passages*: the list of passages of the documents in which the answer to the question is suspected to be

- *Answer-QLF*: a syntactic/quasi-logical representation of the answer (Gaizauskas and Humphreys, 1997)
- *Answer*: the final answer to the question

This data model embodies the two information flows (i.e. question status and related documents) as depicted in Fig. 1.

It is worth noticing that, as not all this information is relevant for all the agents, it should the be easy for each agent to neglect un-relevant information. The access and the data encapsulation is guaranteed by the use of a object oriented programming paradigm over an XML layer for the communication.

Moreover, this *question session* is enough flexible to support different strategies of analysis. For instance, the *query* for the information retrieval engine may be defined at different levels of interpretation and on different level of expansion. A plain query may contains the keywords directly found in the input question whereas an expanded query may contain synonyms of relevant keywords in the query. As discussed in Sec. 3., the role of the expanded query may be appreciated in an activity such as the paragraph re-ranking that is not the document extraction using the information retrieval engine.

2.2. Agent roles: a master/slave approach

The agents in the community have to cooperate in order to achieve the final goal of answering the input question carrying out the activities depicted in Fig. 1. The easiest way to translate this pool of activities in a pool of agents is assigning an activity to each agent. However, as the agents have to cooperate, implementing it in a distributed way the control strategy may be cumbersome. We preferred then a master/slave approach in which the master agent has the control of the solution strategy and of the flowing of the question session. On the other hand, the slave agents will perform a specific task when prompted. It is worth noticing that the specific slave agents are not strictly dependent on the control strategy. This opens the architecture to an easy reconfiguration as the strategy of answering questions may be varied changing the behaviour of the master agent.

3. A case study: A shallow Q&A system

The proposed general model has been tested building a shallow Q&A system consisting of a *question analysis* agent, an *information retrieval* agent, a *passage re-ranking* agent, and an *answer extraction* agent. The model we have designed is triggered by the extraction of four different kinds of information out of the input question:

- the "focus" that "is a phrase in the question that disambiguates it and emphasizes the type of answer being expected"
- the "answer type" that defines the semantic nature of the expected answer
- the "expanded query" obtained on the basis of the input question via expansion using lexical resources as WordNet (Miller, 1995) and FrameNet (Baker et al., 1998).

The expansion of the query is mainly used to re-rank paragraphs before the final answer extraction. The re-ranking model is presented in Sec. 3.1.. On the other hand, the simple model for extracting the answers is discussed in Sec. 3.2..

3.1. Using query expansion for passage re-ranking

The idea of using the expanded query for the passage re-ranking is based on the observation that, at the document level, words appearing in the question may appear as they are but his phenomenon is not so neat in the passage/paragraph in which the answer appears. Using a re-ranking method based on word equivalence classes is a step towards the use of more semantic-aware language interpretation model.

The expanded query Q may be seen as a logical *and* of equivalent classes whose elements are put in an *or* relation, i.e.:

$$Q = C_1 \wedge ... \wedge C_n \quad (1)$$

where C_i is an equivalence class of words represented as $C_i = w_i^1 \vee ... \vee w_i^m$.

In order to describe the model and the re-ranking weight we will use an example, let us focus on the classical question: "Who is the president of United States of America?". In this case, the answer type will be a person and *the president* is the detected focus. The query expansion is done may be done on the relevant words such as *president* and *United States of America*.

In this case a possible expanded query should take into account the possible synonymies for the word *president* (set C_1) and for the word *United States of America* (set C_2). Using WordNet, this is the possible expansion for president C_1= { *president, chief executive, chair, chairperson, chairman, chairwoman, ...* } whereas a possible expansion for *United States of America* is C_2={ *United States, United States of America, America, US, U.S., USA, U.S.A.* }.

The different paragraphs are then ranked according to three hints:

- the relative frequency of the equivalence classes appering in the selected paragraph with respect to the overall number of equivalence classes in the query
- the deviation of the paragraph length from the optimal paragraph lenght
- the *segment* length where a segment is the paragraph portion that encompasses the active elements of an expanded question

The final score is obtained via a composition of these contributes.

3.2. Answer extraction and named entity categories

When the paragraph is selected, a named entities recogniser is applied. Named entities consistent with the answer type are retained as possible answers. The answer type is, whenever possible, obtained using the question focus. Among all the possible answers we select the one that is nearer to the elements of the question.

System Test	
correct answer	13%
answer in the top-10 paragraphs	39%
answer recognized in the top-10 paragraphs	33%
answer in the first paragraph	17%
answer recognized in the first paragraph	71%

Passage Re-renking + Shallow Answer Extractor Test	
correct answer	38%
answer in the top-10 paragraphs	88%
answer recognized in the top-10 paragraphs	43%
answer in the first paragraph	47%
answer recognized in the first paragraph	68%

Table 1: Experimental investigation

3.3. Performance analysis

The performances of the shallow processing model have been investigated on 100 questions of the TREC-2002 on the ACQUAINT collection. Two sets of tests have been carried out in order to evaluate the accuracy of both the full light processing chain and the specific answer extraction module. The main difference between the two tests is in the use of only correct documents in the second test.

The results are shown in Tab. 1 where two different experiments are reported: the results of the overall system (called *System Test*) and the results of the final steps of the model (i.e. *Passage Re-ranking + Shallow Answer Extractor Test*) that aims to investigate the behaviour of the passage re-ranking module. In this second test only documents containing the answer have been retained. We report the measures of how many correct answers have been selected by the system as first answer. This gives the possibility of understanding the overall performances of the system. Moreover, we are interested on how the re-ranking module is working with respect to the possibility of selecting the paragraphs containing the correct answer (reported as the *answer in top-10 paragraphs* and *answer in the first paragraph*) and on how it is able to select with the simple heuristic the correct answer among all the possible ones (reported as the *answer recognized in the top-10 paragraphs* and *answer recognized in the first paragraph*).

Although non state-of-art the above results are interesting especially in the last line. The light system extracts correct answer (about 70%) if first paragraph is the correct paragraph (i.e. it holds the correct answer). As only 17% (47%) of the first paragraphs hold the correct answer we need to improve most of the pre-processing phases, i.e. question espansion (for example, by applying word sense disambiguation before expansion), as well as document retrieval and paragraph selection.

4. Conclusions

Given the agent-based nature of the A2Q system, specific independent measures and extension will be easily applied to the corresponding software components without any of these changes affect other components. Such an algorithmic independence that the system architecture implements will also facilitate the embedding of specific control policies: failures in answer retrieval for the light strategies will be used to trigger deeper techniques and knowledge intensive processes are decided to be applied. This is a typical task for the ”activity scheduler” of the system. Finally multilingual capabilites (the system is able to currently apply an English as well as an Italian robust parser to incoming questions and paragraphs) will be accessed ”on demand” by the scheduler according to the data abstraction represented into the ”question session” in a transparent way for each language dependent process: the design and scalability of language specific components (e.g. the parsers) is thus make independent from the rest of the system. In a field where the assessment of techniques, systems and even general models is still very low, the advantages of the proposed architecture have a strong impact not only on a specific system (e.g. A2Q) but on the overall advances of the Q&A area.

5. References

Baker, Collin F., Charles J. Fillmore, and John B. Lowe, 1998. The berkeley framenet project. In *Proceedings of the COLING-ACL*. Montreal, Canada.

Basili, Roberto and Fabio Massimo Zanzotto, 2002. Parsing engineering and empirical robustness. *Natural Language Engineering*, to appear.

Bellifemine, Fabio, Agostino Poggi, and Giovanni Rimassa, 1999. Jade: A fipa-compliant agent framework. In *Proceedings of PAAM'99*. London.

Gaizauskas, Robert and Kevin Humphreys, 1997. Using a semantic network for information extraction. *Natural Language Engineering*, 3, Parts 2 & 3:147–169.

Miller, George A., 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

MEAD - a platform for multidocument multilingual text summarization

Dragomir Radev[1], Timothy Allison[1], Sasha Blair-Goldensohn[2], John Blitzer[3], Arda Çelebi[4], Stanko Dimitrov[1], Elliott Drabek[5], Ali Hakim[1], Wai Lam[6], Danyu Liu[7], Jahna Otterbacher[1], Hong Qi[1], Horacio Saggion[8], Simone Teufel[9], Michael Topper[1], Adam Winkel[1], Zhu Zhang[1]

[1]University of Michigan, [2]Columbia University, [3]University of Pennsylvania, [4]USC/ISI, [5]Johns Hopkins University, [6]Chinese University of Hong Kong, [7]University of Alabama, [8]University of Sheffield, [9]University of Cambridge
radev@umich.edu

Abstract

This paper describes the functionality of MEAD, a comprehensive, public domain, open source, multidocument multilingual summarization environment that has been thus far downloaded by more than 500 organizations. MEAD has been used in a variety of summarization applications ranging from summarization for mobile devices to Web page summarization within a search engine and to novelty detection.

1. Introduction

MEAD is the most elaborate publicly available platform for multi-lingual summarization and evaluation. Its source and documentation can be downloaded from `http://www.summarization.com/mead`. The platform implements multiple summarization algorithms (at arbitrary compression rates) such as position-based, centroid-based, largest common subsequence, and keywords. The methods for evaluating the quality of the summaries are both intrinsic (such as percent agreement, cosine similarity, and relative utility) and extrinsic (document rank for information retrieval).

MEAD implements a battery of summarization algorithms, including baselines (lead-based and random) as well as centroid-based and query-based methods. Its flexible architecture makes it possible to implement arbitrary algorithms in a standardized framework. Support is provided for trainable summarization (using Decision trees, Support Vector Machines or Maximum Entropy). Finally, MEAD has been used in numerous applications, ranging from summarization for mobile devices to Web page summarization within a search engine and to novelty detection.

2. Architecture

MEAD's architecture consists of four stages. First, documents in a cluster are converted to MEAD's internal (XML-based) format. Second, given a configuration file (.meadrc) or command-line options, a number of features are extracted for each sentence of the cluster. Third, these features are combined into a composite score for each sentence. Fourth, these scores can be further refined after considering possible cross-sentence dependencies (e.g., repeated sentences, chronological ordering, source preferences, etc.) In addition to a number of command-line utilities, MEAD provides a Perl API which lets external programs access its internal libraries. A sample .meadrc file is shown in Figure 1.

All data in MEAD is stored as XML. The following DTDs are part of MEAD:

- cluster: a description of all related documents that will be summarized together,

```
compression_basis sentences
compression_absolute 1
classifier
/clair4/projects/mead307/source/mead/bin/default-classifier.pl
    Centroid 3.0 Position 1.0 Length 15 SimWithFirst 2.0
reranker /clair4/projects/mead307/source/mead/bin/default-reranker.pl
    MEAD-cosine 0.9 enidf
```

Figure 1: Sample .meadrc file. Using this configuration file, MEAD will produce a one-sentence summary using a linear combination of three features as the scoring function. From any sentence pair where the IDF-modified cosine similarity is higher than 0.9, one of the sentences will be dropped.

- docjudge: relevance judgements associated with the document or summary and a particular query and retrieval method,
- docpos: a part-of-speech annotated version of the document,
- docsent: a document, split into sentences,
- document: the raw document,
- extract: alisting of all sentence that should be in the summary,
- mead-config: MEAD's configuration parameters,
- query: a TREC-style query converted to XML,
- reranker-info: parameters for the rerankers,
- sentalign: a sentence-to-sentence alignment across languages,
- sentfeature: a list of feature values for a given document and feature names,
- sentjudge: manually annotated sentences for relevance within a cluster,
- sentrel: CST-style sentence-to-sentence relationships.

A few sample files conforming to these DTDs are shown in the Appendix.

3. Features

The following features are provided with MEAD. They are all computed on a sentence-by-sentence basis.

- Centroid: cosine overlap with the centroid vector of the cluster (Radev et al., 2004),
- SimWithFirst: cosine overlap with the first sentence in the document (or with the title, if it exists),
- Length: 1 if the length of the sentence is above a given threshold and 0 otherwise,
- RealLength: the length of the sentence in words,
- Position: the position of the sentence in the document,
- QueryOverlap: cosine overlap with a query sentence or phrase,
- KeyWordMatch: full match from a list of keywords,
- LexPageRank: eigenvector centrality of the sentence on the lexical connectivity matrix with a defined threshold.

4. Classifiers

Four classifiers come with MEAD.

- Default: provides a linear combination of all features except for "Length" which is treated as a cutoff feature (see previous section),
- Lead-based: a baseline classifier that favors sentences that appear earlier in the cluster, as defined by the order of documents in the definition of the cluster,
- Random: a baseline classifier that extracts sentences at random from the cluster,
- Decision-tree: a machine learning algorithm, based on Weka (Witten and Frank, 2000) and trained on an annotated summary corpus.

5. Rerankers

The following rerankers are included in MEAD.

- Identity: this reranker does nothing; it preserves the scores of all sentences as computed by the classifier,
- Default: keep all scores, but skip sentences that are too similar (cosine similarity above a specific threshold) to sentence already included in the summary,
- Time-based: penalize earlier (or later, depending on the argument) sentences,
- Source-based: penalize sentences that come from particular sources,
- CST-based: this reranker applies different policies as determined by the cross-document structure of the cluster (Radev, 2000; Zhang et al., 2002),
- Maximal Marginal Relevance (MMR): this reranker is based on the MMR principle as formulated in (Carbonell and Goldstein, 1998).

6. Evaluation methods

The MEAD evaluation toolkit (MEADEval), previously available as a separate piece of software, has been merged into MEAD as of version 3.07. This toolkit allows evaluation of human-human, human-computer, and computer-computer agreement. MEADEval currently supports two general classes of evaluation metrics: co-selection and content-based metrics. Co-selection metrics include precision, recall, Kappa, and Relative Utility, a more flexible cousin of Kappa. MEAD's content-based metrics are cosine (which uses TF*IDF), simple cosine (which doesn't), and unigram- and bigram-overlap. An additional metric, relevance correlation, is available as an addon.

- Precision/recall: which sentences in the summary match the sentences in the human model,
- Kappa: takes into account interjudge agreement as well as the difficulty of the problem,
- Relative utility: similar to Kappa but allows for non-binary judgements in the model,
- Relevance correlation: there are two versions of this metric: Spearman (rank correlation) and Pearson (linear correlation); given a query, a search engine, and a document collection, Relevance correlation is high if a ranked list of the full documents in the collection given the query is highly correlated with a similar rankings based on the summaries of the documents.
- Cosine: cosine similarity against a human summary (or a set of human summaries),
- Longest-common subsequence: same as Cosine, but using the longest-common subsequence similarity measure,
- Word overlap: same as Cosine, but based on the number of words in common between the automatic and manual summaries,
- BLEU: based on the precision-oriented n-gram matcher developed by (Papineni et al., 2002).

7. Corpora

- SummBank: this is a large corpus for summary evaluation. It CD-ROM contains 40 news clusters in English and Chinese, 360 multi-document, human-written non-extractive summaries, and nearly 2 million single document and multi-document extracts created by automatic and manual methods. The collection was prepared as part of the 2001 Johns Hopkins summer workshop on Text Summarization (Radev et al., 2002).
- CSTBank: a smaller corpus, manually annotated at the University of Michigan for CST (Cross-document Structure Theory) relationships. CST relationships include subsumption, identity, fulfillment, paraphrase, elaboration/refinement, etc.

8. Utilities

The following utilities are included in MEAD:

- DUC conversion: scripts to convert DUC 2002–2004 style SGML documents into the MEAD format,
- Sentjudge to manual summary conversion: scripts to generate manual summaries from manual sentence-based non-binary relevance judgements,
- CIDR: a document clustering utility partially built over the MEAD API,
- Preprocessors: tools to convert plain text and HTML documents to the MEAD format.
- Sentrel utilities: tools to manipulate CST-style sentence relevance judgements.

9. Applications

MEAD has been successfully used in the following tasks: evaluate an existing summarizer, test a summarization feature, test a new evaluation metric, test a short-query machine translation system. It has also been used in major evaluations such as DUC (Radev et al., 2001a; Otterbacher et al., 2002; Radev et al., 2003) (text summarization) and TREC (question answering and novelty detection). Several systems have been built on top of MEAD, specifically NewsInEssence(Radev et al., 2001c; Radev et al., 2001b) (online news tracking and summarization), WebInEssence (Radev et al., 2001d) (clustering and summarization of Web hits), and WAPMead (in progress) (wireless access to summarization for email access).

10. History

MEAD v1.0 and v2.0 were developed at the University of Michigan in 2000 and early 2001. MEAD v3.01 – v3.06 were written in the summer of 2001 at Johns Hopkins University. As of Version 3.07, MEAD has been back to Michigan. The current release, 3.07, includes support for English and Chinese in a UNIX (Linux/Solaris/Cygwin) environment. Adding new (human) languages should be equally easy.

11. Acknowledgments

This work was partially supported by the National Science Foundation's Information Technology Research program (ITR) under grant IIS-0082884. All opinions, findings, conclusions and recommendations in any material resulting from this workshop are those of the participants, and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank a number of individuals for making MEAD possible: Fred Jelinek, Sanjeev Khudanpur, Laura Graham, Naomi Daniel, Anna Osepayshvili, and many others.

Appendix. Sample XML files

The following figures: 2, 3, 4, 5, 6, and 7 give illustrations of various XML files used by MEAD.

```
<?xml version='1.0'?>

<SENT-FEATURE>
<S DID="87" SNO="1" >
<FEATURE N="Centroid" V="0.2749" />
</S>
<S DID="87" SNO="2" >
<FEATURE N="Centroid" V="0.8288" />
</S>
<S DID="81" SNO="1" >
<FEATURE N="Centroid" V="0.1538" />
</S>
<S DID="81" SNO="2" >
<FEATURE N="Centroid" V="1.0000" />
</S>
<S DID="41" SNO="1" >
<FEATURE N="Centroid" V="0.1539" />
</S>
<S DID="41" SNO="2" >
<FEATURE N="Centroid" V="0.9820" />
</S>
</SENT-FEATURE>
```

Figure 2: Sentfeature object

```
<?xml version='1.0'?>
<SENT-JUDGE QID='551'>
<S DID='D-19980731_003.e' PAR='1' RSNT='1' SNO='1'>
                <JUDGE N='smith' UTIL='10'/>
                <JUDGE N='huang' UTIL='10'/>
                <JUDGE N='moorthy' UTIL='6'/>
</S>
<S DID='D-19980731_003.e' PAR='2' RSNT='1' SNO='2'>
                <JUDGE N='smith' UTIL='6'/>
                <JUDGE N='huang' UTIL='10'/>
                <JUDGE N='moorthy' UTIL='10'/>
</S>
<S DID='D-19980731_003.e' PAR='3' RSNT='1' SNO='3'>
                <JUDGE N='smith' UTIL='6'/>
                <JUDGE N='huang' UTIL='9'/>
                <JUDGE N='moorthy' UTIL='10'/>
</S>
<S DID='D-19981105_011.e' PAR='5' RSNT='2' SNO='7'>
                <JUDGE N='smith' UTIL='2'/>
                <JUDGE N='huang' UTIL='1'/>
                <JUDGE N='moorthy' UTIL='4'/>
</S>
</SENT-JUDGE>
```

Figure 3: Sentjudge object

```
<?xml version='1.0'?>
<!DOCTYPE DOC-JUDGE SYSTEM '/clair4/mead/dtd/docjudge.dtd'>

<DOC-JUDGE QID='Q-2-E' SYSTEM='SMART' LANG='ENG'>
    <D DID='D-19981007_018.e' RANK='1' SCORE='9.0000' />
    <D DID='D-19980925_013.e' RANK='2' SCORE='8.0000' />
    <D DID='D-20000308_013.e' RANK='3' SCORE='7.0000' />
    <D DID='D-19990517_005.e' RANK='4' SCORE='6.0000' />
    <D DID='D-19981017_015.e' RANK='4' SCORE='6.0000' />
    <D DID='D-19990107_019.e' RANK='12' SCORE='5.0000' />
    <D DID='D-19990713_010.e' RANK='12' SCORE='5.0000' />
    <D DID='D-19991207_006.e' RANK='12' SCORE='5.0000' />
    <D DID='D-19990913_001.e' RANK='20' SCORE='4.0000' />
    <D DID='D-19980609_005.e' RANK='20' SCORE='4.0000' />
    <D DID='D-19990825_018.e' RANK='1962' SCORE='0.0000' />
    <D DID='D-19990924_047.e' RANK='1962' SCORE='0.0000' />
</DOC-JUDGE>
```

Figure 5: Docjudge object

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE EXTRACT SYSTEM '/clair/tools/mead/dtd/extract.dtd'>

<EXTRACT QID='GA3' LANG='ENG' COMPRESSION='7'
SYSTEM='MEADORIG' RUN='Sun Oct 13 11:01:19 2002'>
<S ORDER='1' DID='41' SNO='2' />
<S ORDER='2' DID='41' SNO='3' />
<S ORDER='3' DID='41' SNO='11' />
<S ORDER='4' DID='81' SNO='3' />
<S ORDER='5' DID='81' SNO='7' />
<S ORDER='6' DID='87' SNO='2' />
<S ORDER='7' DID='87' SNO='3' />
</EXTRACT>
```

Figure 7: Extract Object

```
<?xml version='1.0'?>
<!DOCTYPE QUERY SYSTEM "/clair4/mead/dtd/query.dtd" >

<QUERY QID="Q-551-E" QNO="551" TRANSLATED="NO">
<TITLE>
Natural disaster victims aided
</TITLE>
<DESCRIPTION>
The description is usually a few sentences describing the cluster.
</DESCRIPTION>
<NARRATIVE>
The narrative often describes exactly what the user is looking for in the summary.
</NARRATIVE>
</QUERY>
```

Figure 4: Query object

```
<MEAD-CONFIG TARGET='GA3' LANG='ENG' CLUSTER-PATH='/clair4/mead/data/GA3'
  DATA-DIRECTORY='/clair4/mead/data/GA3/docsent'>

<FEATURE-SET BASE-DIRECTORY='/clair4/mead/data/GA3/feature/'>
  <FEATURE NAME='Centroid' SCRIPT='/clair4/mead/bin/feature-scripts/Centroid.pl HK-WORD-enidf ENG'/>
  <FEATURE NAME='Position' SCRIPT='/clair4/mead/bin/feature-scripts/Position.pl'/>
  <FEATURE NAME='Length' SCRIPT='/clair4/mead/bin/feature-scripts/Length.pl'/>
</FEATURE-SET>

<CLASSIFIER COMMAND-LINE='/clair4/mead/bin/default-classifier.pl \
  Centroid 1 Position 1 Length 9' SYSTEM='MEADORIG' RUN='10/09'/>

<RERANKER COMMAND-LINE='/clair4/mead/bin/default-reranker.pl MEAD-cosine 0.7'/>

<COMPRESSION BASIS='sentences' PERCENT='20'/>

</MEAD-CONFIG>
```

Figure 6: Mead-config object

12. References

Carbonell, Jaime G. and Jade Goldstein, 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In Alistair Moffat and Justin Zobel (eds.), *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Melbourne, Australia.

Otterbacher, Jahna C., Dragomir R. Radev, and Airong Luo, 2002. Revisions that improve cohesion in multi-document summaries: a preliminary study. In *Proceedings of the Workshop on Automatic Summarization (including DUC 2002)*. Philadelphia: Association for Computational Linguistics.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Radev, Dragomir, 2000. A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *Proceedings, 1st ACL SIGDIAL Workshop on Discourse and Dialogue*. Hong Kong.

Radev, Dragomir, Sasha Blair-Goldensohn, and Zhu Zhang, 2001a. Experiments in single and multi-document summarization using MEAD. In *First Document Understanding Conference*. New Orleans, LA.

Radev, Dragomir, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Arda Çelebi, Hong Qi, Elliott Drabek, and Danyu Liu, 2002. Evaluation of text summarization in a cross-lingual information retrieval framework. Technical report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD.

Radev, Dragomir R., Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan, 2001b. Interactive, domain-independent identification and summarization of topically related news articles. In *5th European Conference on Research and Advanced Technology for Digital Libraries*. Darmstadt, Germany.

Radev, Dragomir R., Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan, 2001c. Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Human Language Technology Conference (Demo Session)*. San Diego, CA.

Radev, Dragomir R., Weiguo Fan, and Zhu Zhang, 2001d. Webinessence: A personalized web-based multi-document summarization and recommendation system. In *NAACL Workshop on Automatic Summarization*. Pittsburgh, PA.

Radev, Dragomir R., Hongyan Jing, Malgorzata Stys, and Daniel Tam, 2004. Centroid-based summarization of multiple documents. *Information Processing and Management, in press*.

Radev, Dragomir R., Jahna Otterbacher, Hong Qi, and Daniel Tam, 2003. Mead reducs: Michigan at duc 2003. In *Proceedings of DUC 2003*. Edmonton, AB, Canada.

Witten, Ian H. and Eibe Frank, 2000. *Data Mining: Practical machine learning tools with Java implementations*. San Francisco: Morgan Kaufmann.

Zhang, Zhu, Sasha Blair-Goldensohn, and Dragomir Radev, 2002. Towards CST-enhanced summarization. In *Proceedings of the AAAI 2002 Conference*. Edmonton, Alberta.

NLP-enhanced Content Filtering within the POESIA Project

Mark Hepple[⋆], Neil Ireson[⋆], Paolo Allegrini[†], Simone Marchi[†], Simonetta Montemagni[†] and Jose Maria Gomez Hidalgo[◇]

[⋆]University of Sheffield, Department of Computer Science, Regent Court, 211 Portobello Street, Sheffield, UK. {m.hepple, n.ireson}@dcs.shef.ac.uk

[†]Istituto di Linguistica Computazionale, CNR, Area della Ricerca di Pisa Via Moruzzi 1, 56124 Pisa, Italy. {allegrip, simone.marchi, simonetta.montemagni}@ilc.cnr.it

[◇]Departamento de Inteligencia Artificial, Universidad Europea de Madrid, 28670, Villaviciosa de Odon, Madrid, Spain. jmgomez@uem.es

Abstract

This paper introduces the POESIA internet filtering system, which is open-source, and which combines standard filtering methods, such as positive/negative URL lists, with more advanced techniques, such as image processing and NLP-enhanced text filtering. The description here focusses on components providing textual content filtering for three European languages (English, Italian and Spanish), employing NLP methods to enhance performance. We address also the acquisition of language data needed to develop these filters, and the evaluation of the system and its components.

1. Introduction

POESIA (Public, Open-source Environment for Safer Internet Access: IAP 2117/27572) is a multisite project funded under the EU Internet Action Plan, which is developing an advanced internet filtering system, intended primarily for use in schools and other educational establishments, with the aim of providing safe and educationally appropriate internet access for young people. The system is *open-source*, providing a basis for its development and maintenance beyond the project's lifetime, and is freely available for download and installation.[1] In this paper, we will sketch the overall POESIA system, and then provide greater detail of the methods used for filtering on the basis of textual content. Considerable quantities of web-page data are required both for the development and evaluation of the system. We will describe the approach taken for collecting this data, and the available early results on evaluation.

2. The POESIA System

POESIA's approach is to use multiple filters, each of which addresses some source of evidence that is of potential use in identifying harmful pages. The evidence detected can then be combined by a Decision Mechanism (DM) component to produce an overall decision for each page. In this way, POESIA can best exploit whatever information is available to determine whether pages should be filtered. Work to date on the system has focussed on the filtering of pornographic content, but the same mechanisms could be reapplied to other domains.

At the heart of the POESIA architecture is a central controller, the Monitor, which interfaces with the internet caching proxy (e.g. Squid or Shweby), or other filtering client, to determine the pages that must be assessed for filtering, and to return accept/reject decisions. The Monitor also invokes the other system components, and facilitates the traffic of data and results between filters and the DM, and caches filtering results for recently seen pages.

The POESIA filters include some that implement widely used filtering methods, e.g. positive/negative URL lists and PICS.[2] These methods are fairly effective for the sites and pages explicitly addressed, but given the enormous size of the internet and its dynamic character, these approaches can only ever achieve partial coverage. This suggests the need for filtering based on the *content* of pages, and POESIA includes filters addressing both image and textual content. The image filter assesses the likelihood that the images within a page are pornographic, based on the proportion of image area corresponding to skin, and on the shape and orientational characteristics of major skin areas. The multiple filters of the system should be seen as operating in combination. For example, a page from a site which is not on the URL lists will be analysed for content. If the page contains a reasonable quantity of text, this alone might allow a reject decision, but if there is limited text, it might require the combination of image and text evidence for a decision to be made. The DM plays a crucial role in weighing the available evidence to produce an overall decision.

The POESIA architecture readily allows for the inclusion of additional or substitute filters, and so the open-source character of the project allows for the continuing development and relevance of the system into the future.

3. Filtering for Textual Content

Within POESIA, three language-specific text filters have been developed by different sites which specialise in

[1]See `http://www.poesia-filter.org` for a full listing of the project partners, additional information on the system, and a link to the open-source repository from which the system can be downloaded.

[2]PICS (Platform for Internet Content Selection) is a scheme by which web content providers can assign labels rating the content of their pages, which can be used directly by a suitably configured browser to prevent children accessing pages with inappropriate content.

NLP for the target languages, which are English, Italian and Spanish. The filters differ in the methods they employ, partly reflecting an attempt to optimise over the different NLP resources available for each language. However, the filters are alike in offering both 'light' and 'heavy' filtering modes. Light filtering, which uses little NLP, provides rapid assessment of content for straightforwardly classifiable pages. For other pages, heavy filtering, making greater use of NLP, is invoked to provide more sensitive detection of content indicators. This trade-off is important to the overall efficiency of the system. In the case of pages that contain insufficient text for a conclusion to be drawn, filters can return a special result unknown.

The POESIA system includes a language identifier component, which is required to ensure that page text is routed to the appropriate language-specific text filter. This component uses a standard approach based on character n-gram statistics, see e.g. (Cavnar & Trenkle, 1994)).

4. Data Acquisition

The language identifier was trained using a large ($\sim$560Mbytes), publicly-available, parallel corpus, which covers 11 European languages, including English, French, Italian and Spanish.

The development and testing of the POESIA text filters requires a substantial quantity of pages for each language, which have been precategorised as pornographic and non-pornographic. Manual collection of this data would be infeasible. Instead, the data was automatically spidered from the WWW, using the Google directory structure (http://directory.google.com) to locate sites which fall into the pornographic or non-pornographic category. The spider tranverses links within identified sites to retrieve pages at varying depths. Pages are stripped of HTML and the text is analysed to ensure it is of the target language and to highlight potential misclassifications. Despite these checks, there will inevitably be some number of pages which are incorrectly classified, and this fact will be reflected in the final performance scores. The corpus collected for each language ranges in size between 5k and 20k pages.

5. Text Filtering for English

The English light filter employs a conventional statistical approach to text classification, using a bag-of-words representation, with stoplisting and stemming. Indexing terms are selected via a minimum threshold for document frequency in the training corpus. A model is constructed of each category consisting of a ranked frequency list of index terms. Classification is done using an out-of-place measure over term frequency rankings.

The English heavy filter focuses on pages that have been misclassified as non-pornographic by the light filter during training. A set of keywords is identified from these pages, which are the n highest-ranking terms according to the tf.idf measure. A value of n=10 was found to be suitable in this context. As might intuitively be expected, these terms commonly appear to be indirect indicators of pornographic content, e.g. *adult*, *explicit*. An instance-based approach is used to learn contextual differences for the use of these keywords between pages that have been correctly and incorrectly classified as pornographic by the light filter. The contextual pattern is determined by a window of words around the keyword. The learning process can generalise these patterns by replacing words with their stem, POS tag or a named entity (NE) label, or with a "wildcard" symbol. The pattern matching process can either consider the absolute position of the adjacent words with respect to the keyword, or consider the preceeding and following words as an ordered list or unordered set. The best predictive patterns were produced by a 6-word window: a smaller window did not provide enough context to differentiate keyword use and a larger window did not improve the prediction. In addition, our experiments showed that no significant benefit resulted from allowing generalisation during learning to either POS tags or to NE categories of the kind produced by standard NE recognition systems, i.e. person, company, date, etc. However, benefit was found for generalisation by stemming and by a special case of NE recognition in which person names are categorised for gender. An approach of representing contexts as a list was found to perform better than one representing them as a set. At runtime, any documents that are classified as non-pornographic by the light filter, but which contain keyword occurrences, are passed to the heavy filter which applies the contextual patterns to determine the predicted document class.

The underlying approach of the English heavy filter can be seen as one of using local contextual cues to disambiguate between alternative uses or senses of key terms, as is relevant to particular categorisations. As such, the approach can be likened to that of (Riloff & Lorenzen, 1999), except that they use a linguistically richer representation, for example including aspects of syntactic structure. The simpler approach we have used has potential advantages in terms of portability (i.e. to other domains and languages), and robust application, since the contexts in which the keywords appear in html-stripped web pages may not correspond to grammatical sentences, and yet may exhibit regularities facilitating category prediction.

6. Text Filtering for Italian

The Italian light filter works at two levels. The first level employs a statistical word-based categorization, using local term counts rather than global frequencies. Text is tokenised and segmented into windows of 100 words. Each window is assigned a score based on the maximum local frequency of domain relevant words (markers). For each text, the filter outputs the maximum cumulative word score over different text windows. For efficiency, given the morphological richness of Italian, the morphological variants of $\sim$40 unambiguous marker lemmata, extracted from a linguistically annotated training courpus, are precomputed. The second level consists of recognition of relevant regular expressions, extracted from the training corpus, mostly associated with warnings (e.g. "adult content", or "download the dialler program"), with all possible lexical variations. Even though this recognition is implemented in the light filter, the identification of these expressions has required the use of some advanced NLP techniques, for the extraction of multi-word terms and the detection of semantic similar-

ity. Thresholds map text scores to low/medium/high values; low/high values are notified directly to the DM. For medium values, the heavy filter is invoked.

The heavy filter operates on morpho-syntactically tagged and lemmatized texts. For this purpose, we used a tool combining ILC's morphological analyzer MAGIC, and an optimized version of the Brill tagger. Filtering is based on recognition of ~ 2400 domain relevant lemmata (including ambiguous words). Category learning uses an entropy-based classifier: CASSANDRA (Complex Analysis of Sequences via Scaling AND Randomness Assessment), which computes the rate of information increase generated by salient lemmata. Shannon's information S for the probability $P(x;l)$ of finding a fixed number x of "salient" lemmata in a moving window of length l was recently shown to give a maximal entropy change $dS/d(\log l)$, when genre-salient lemmata are selected (Allegrini, et al., 2004). A major role is played by the concept of scaling, defined by

$$p(x,l) = \frac{1}{l^{\delta}} F(\frac{x}{l^{\delta}}). \quad (1)$$

Complex systems, obeying Zipf's law, are expected to generate a departure from the condition of ordinary statistics, where $\delta = 0.5$ and $F(y)$ is a Gaussian function of y. The computation of the Shannon's information functional, in the case when the property of Eq. (1) applies, is easily proved to yield $S(l) = A + \delta \ln(l)$, where A is a constant whose explicit expression is of no interest here. It is evident that with this method the scaling parameter is easily evaluated by plotting $S(l)$ in a linear-log representation.

The CASSANDRA method works as follows. We study a time series that is not stationary. Then, we supplement the Shannon entropy method with the introduction of a big window of size L, which has to be considered as a sequence of its own, and we move it along the sequence being analysed, for the purpose of assessing its local properties. The size of this window has to be large enough as to make it possible to make a statistical analysis (in practice, we choose $L = 100$ words). For some positions of the big window we evaluate the quantity

$$C_j(\lambda) = \frac{\sum_{l=1}^{\lambda} [S_j(l) - S_j(1) - 0.5 \ln(l)]}{\lambda} \quad (2)$$

where $S_j(l)$ and $S_j(1)$ denote the Shannon entropies corresponding to small windows of size l and 1, respectively, moving within the big window with position j. Eq. (2) means comparing the actual entropy change to the ideal change occurring with an infinitely fast (Poissonian) transition to randomness, corresponding, to the entropy increasing as $0.5 \ln(l)$. The validity of Eq. (2) rests on the mathematical inequality $\lambda << L << N$. With the condition $L << N$, we can locate the big window in different positions of the text, identifying *where* the domain relevant (i.e. erotic) lemmata are meaningful, with a large information increase. The condition $\lambda << L$ makes it possible for us to use enough data to reach a conclusion about the statistical property of the small region under observation. This is why the CASSANDRA classifier is able to perform well in difficult tasks, detecting pages containing *erotic stories* vs, for instance, pages of *sexual education*, making a "wise" use of ambiguous terms. On the other hand, for pages with only a small amount of text, performance does not improve significantly over that of the light filter.

7. Text Filtering for Spanish

The Spanish light filter uses state-of-the-art text categorization techniques (Sebastiani, 2002). Text in Web pages is firstly tokenized, stoplisted, and stemmed. The top 1% Information Gain (IG) scoring terms of the training data are used to represent pages as term-weight vectors according to the Vector Space Model (VSM), using binary weights. A linear Support Vector Machine (SVM) classifier is trained over this representation, to classify new pages as either Porn or non-Porn. The Spanish heavy filter uses the same machine learning approach, but with two additional, linguistically motivated, multi-word input features: Noun Phrases and Named Entities.

- Noun Phrases are recognised via part of speech tagging and regular expression matching according to a compact noun phrase grammar. The part of speech tagger follows a Maximum Entropy approach (Ratnaparkhi, 1998) trained on the Spanish CONLL'02 corpus. The Maximum Entropy tagger has been iterated until an accuracy of 96% is reached on the training collection. The phrases found in the training phase are normalized by stoplist filtering, stemming individual words and alphabetical ordering.
- Secondly, Named Entities in the training collection are recognized using a subset of the attributes suggested in (Carreras et al., 2002), and the decision tree learner C4.5 trained on the CONLL'02 Spanish Corpus. Attributes considered in our approach include the actual words in a 5-word window around the target word, and capitalization properties of these words. The current version reaches a F_1=0.828 on the CONLL'02 test collection when considering only Named Entities but not their type (locations, persons, organizations and miscelanea). Again, Named Entities are normalized as Noun Phrases.

Named Entities and Noun Phrases are taken as additional features to stoplisted, stemmed words in a VSM binary representation. We retain the 10% top IG scoring features, and learn a linear SVM classifier over the training collection, as with the light filter. The evaluation of this approach has not yet been completed, but we believe these additional features will improve the effectiveness of learning, producing a more effective, if also more time-consuming, classifier.

8. Results and Discussion

To evaluate the various language-specific text filtering components, we have tested them in direct use as classifiers of pornographic vs. non-pornographic web pages. [3] The re-

[3]The filters normally provide an assessment of content as input to the DM, so this direct use as binary classifiers is not their normal context of use. The English and Italian filters may return a result 'unknown', when the decision for a page is unclear. In the results reported here, such pages are given a default assignment to either the porn (for Italian) or non-porn (for English) category.

ALL TEXTS			Light Filter					Light+Heavy Filters				
Language	Category	Pages	Prec	Rec	F_1	Eff	OvB	Prec	Rec	F_1	Eff	OvB
English	Porn	5090	.969	.938	.953	93.8%	3.2%	.967	.952	.960	95.2%	3.4%
	Non-Porn	4840	.937	.968	.952			.951	.966	.958		
Italian	Porn	3500	.948	.963	.955	96.3%	4.4%	.975	.953	.964	95.3%	2.0%
	Non-Porn	4195	.968	.956	.962			.961	.980	.971		
Spanish	Porn	1000	.995	.916	.953	91.6%	1.9%					
	Non-Porn	4000	.999	.981	.989							

Figure 1: Performance results for text filters

sults are given as precision and recall scores for each category (i.e. porn, non-porn) together with the corresponding F-measure scores (F_1). In addition to these familiar metrics, percentage scores are also given for two additional metrics which are widely used in a filtering context. These are *effectiveness* (Eff), which is the proportion of harmful pages blocked (here corresponding to recall for the porn category), and *overblocking* (OvB), which is the proportion of harmless pages that are incorrectly blocked (equivalent here to one minus recall for non-porn). Results are provided both for light filters alone, and for where the light and heavy filters of a language are used together.

Scores for the combined light/heavy filter for English and Italian indicate benefits for both languages of including the heavy filter, as shown by increased F_1 values. (The corresponding scores for Spanish were not available at the time of completing the paper.) However, the key benefit observed differs between the two languages. For English, we see a reduction in error rate for porn of 22.6%, i.e. so that the number of harmful pages incorrectly allowed through is reduced by nearly a quarter. For Italian, the key benefit is a reduction in overblocking, such that the number of harmless pages that are incorrectly blocked is reduced by $\sim$55%, although this is accompanied by some reduction in the effectiveness score.

Not surprisingly, the performance of text filters is significantly affected by the quantity of text within files. To take the case of English (although similar observations could be made for the other languages), excluding files that contain $\leq$20 distinct terms produces the following results:

$>$20 *terms*	Light filter			Light+Heavy		
Category	Prec	Rec	F_1	Prec	Rec	F_1
Porn	.979	.959	.969	.977	.976	.977
Non-Porn	.959	.979	.969	.976	.977	.976

Comparing to Figure 1, we see for light filtering that the F_1 rises from around .953 to .969 for both porn and non-porn, and for light+heavy filtering, F_1 rises from around .96 to around .977. For these higher content pages, the heavy filter reduces the error (misclassified pages) for porn by over 40%. Performance for the omitted low text content pages is accordingly lower (with F_1's around .90). However, we would expect pornographic pages with low text content to have high image content, and hence to be identified by the image filter, so that the combination of image and text filtering can perform more effectively than either alone. Evaluation of such combined filtering (i.e. image+text) is at a preliminary stage, but early results do suggest that this synergy of content filters does occur.

A question that might be raised regarding the overall POESIA system is whether it will allow for filtering of other languages, given that specialist NLP techniques have been used in the filters developed for the key target languages of English, Italian and Spanish. It should be noted, however, that the light filter systems developed for the three languages all employ generic text classification approaches, that can readily be reused to produce light filters for other languages, provided that a sufficient quantity of categorised training data can be acquired. This portability has been demonstrated within the project by the creation of a light filter for French using the code developed for the English light filter. In addition, the flexible architecture and open-source character of POESIA allow that heavy text filters for additional languages can be incorporated into the system should there by groups willing to develop them.

9. References

Allegrini, P., Grigolini, P. and Palatella, L. (2004). Intermittency and scale-free networks: a dynamical model for human language complexity, *Chaos, Solitons & Fractals*, v. 20, pp. 95-105.

Carreras, X., Márques, L. and Padró, L. (2002). Named Entity Extraction using AdaBoost. In *Proceedings of the Sixth Conference on Computational Natural Language Learning*.

Cavnar, W.B. and Trenkle, J.M. (1994). N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 161–175.

Ratnaparkhi, A. (1998). *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Riloff, E. and Lorenzen, E. (1999). Extraction-based text categorization: Generating domain-specific role relationships automatically. In T. Strzalkowski, ed., *Natural Language Information Retrieval*. Kluwer AP.

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

SESSION P24-T

Terminology Tools & Data

Using Weighted Abduction to Align Term Variant Translations in Bilingual Texts

Michael Carl, Ecaterina Rascu and Johann Haller

Institut für Angewandte Informationsforschung
Martin-Luther-Str. 14, Saarbrücken, Germany
{carl, kati, hans}@iai.uni-sb.de

Abstract

In this paper we describe a method for detecting terminological variants and their translations in bilingual texts. Our approach is based on abductive reasoning and combines various monolingual and bilingual resources. A small scale experiment shows that precision and recall increase when using more resources and when the resources interfere in a less restricted way. In order to tune our system, we develop a weighing strategy based on the precision of term translation alignments in a reference text. We feed these weights back into the linguistic resources and repeat the experiment. The results show that precision values are considerably higher when weighing term alignments.

1. Introduction

The consistent use of terms in technical domains increases the comprehensibility and translatability of texts (Mitamura and Nyberg, 1995). However, terminological variation is a frequent phenomenon even in established domains (Daille et al., 1996; Macklovitch, 1995; Royauté, 1999).

Enguehardt distinguishes between term recognition systems (TRS) and term extraction systems (Enguehard, 2003). While the latter identify new terms in texts, the former ones detect variants of already known terms.

In this paper we investigate an abductive method to detect terms and their translations in bilingual texts. The proposed architecture combines two monolingual term recognition systems capable of identifying terms and their variants. We infer term variation templates from language specific general variation patterns by means of abduction and use them to identify term variant translations in aligned texts. Abduction as "inference to the best explanation" also requires a ranking of the hypotheses by evaluating their explanatory power (Magnani, 2001). We achieve this by weighing term variation templates according to the co-occurrence precision of variation patterns.

We first present the approach adopted for term recognition. In section 3., we evaluate the system in a number of different settings.

2. Abductive Approach to Term Recognition

To detect translations of terms and their variants in an aligned English–French text, the system requires two types of resources. The first resource is a bilingual terminology containing base terms and their authorized translations. The second resource consists of language specific general variation patterns and synonymy relations. Based on these variation patterns and the terminology, a number of term specific variation templates are generated for every term in the bilingual terminology. The variation templates are stored in a database — the so-called Abductive Terminology Database (ATDB) — together with the original terms so that each variant is linked to its authorized form. The architecture is plotted in figure 1. The actual ATDB is in the center of figure 1 and will be discussed in section 2.2..

An ATDB consists of two symmetrical language sides, a left-hand English side and a right-hand French side[1]. A bilingual sentence aligned text is fed into the system which detects term translations and marks them accordingly. The automatically annotated text is then compared with the manually annotated version of the same text. Values for precision and recall are computed for every term and template. We accumulate weights for general variation patterns based on precision values of term templates and feed these weights back into the resources. We refer to this mechanism as weighted abduction. In section 3. we outline this approach in more depth and show how it can be used to grade ambiguities and reduce noise.

In the following subsections we present the different resources of the ATDB in more detail.

2.1. General Term Variation Patterns

We distinguish three types of variations: typographical variations, morpho-syntactic variations and lexical variations. In this section we give examples of these.

2.1.1. Typographical Variation

Typographical variants differ in the way hyphenation, blanks or punctuation marks are used around a term constituent. Examples are given in (1) and (2). We write the authorized term on the left-hand side of the arrow and the variant on the right-hand side.

(1) *hand stop* → *handstop*
(2) *re-insert* → *reinsert*

2.1.2. Morpho-syntactic Variation

Morpho-syntactic variants are derived from a base term by morphological derivation and/or by transformation of its syntactic structure. The basic mechanisms of structural transformation are omission, insertion, permutation, and coordination (Jacquemin, 1996; Daille et al., 1996). Omission implies the deletion of one or more components from

[1] see also (Carl et al., 2004) for a more detailed discussion.