D5.4

Text mining and linking module – v2

v 1.1 / 2017-02-10

Alexandru Ginsca (CEA), Hervé Le Borgne (CEA), Adrian Popescu (CEA), Symeon Papadopoulos (CERTH), Yiannis Kompatsiaris (CERTH)

The current deliverable is a technical report accompanying the second version of the USEMP text mining and linking modules. The primary objective of these modules is to process the text content that is associated with the users of Online Social Networking (OSN) services (e.g. their posts and comments, the content of the articles they like, etc.) in order to extract personal information cues that could be used for user profiling.
In particular, this deliverable documents the underlying principles and methodologies of the developed modules, the exposed functionality, the respective implementation details, and the conducted evaluation experiments. During the second iteration of the project, work focused on three modules, namely multilingual opinion mining, entity linking and text based location recognition.

Project acronym	USEMP
Full title	User Empowerment for Enhanced Online Presence Management
Grant agreement number	611596
Funding scheme	Specific Targeted Research Project (STREP)
Work program topic	Objective ICT-2013.1.7 Future Internet Research Experimentation
Project start date	2013-10-01
Project Duration	36 months

Workpackage	5
Deliverable lead org.	CEA
Deliverable type	Prototype
Authors	Alexandru Ginsca (CEA), Hervé Le Borgne (CEA), Adrian Popescu (CEA), Symeon Papadopoulos (CERTH), Yiannis Kompatsiaris (CERTH)
Reviewers	Georgios Petkos (CERTH) Rob Heyman (iMinds)
Version	1.1
Status	Final
Dissemination level	RE: Restricted Group
Due date	2015-11-30
Delivery date	2016-05-13 (revised 2017-02-10)

Version	Changes
0.1	ToC from CEA
0.2	First version of location detection module inserted
0.3	Location detection module description updated
0.4	Opinion mining module presentation added
0.5	Entity linking module description included
0.6	Refinements after first internal review
1.0	Refinements after second internal review
1.1	Updated version addressing comments received from the third annual review

Table of Contents

1.Introduction

This deliverable provides a description of the USEMP opinion mining, entity linking and location detection modules implemented during the second iteration of the project. The introduction gives an overview of the role of text mining in USEMP, of the research methodology and of multidisciplinary interactions within the project.

The main objectives of the deliverable are:

a) to extend the functionalities of text mining modules in the USEMP framework;
b) to detail the research approaches adopted, including implementation details;
c) to present an evaluation of the improved and new text mining modules.

The main objective of text mining and linking modules in USEMP is to give the system the capability to conduct inferences about OSN users' interests and traits based on the content of the texts they share and interact with. Inferences are produced for individual texts and are subsequently shown in different parts of Databait:

- Independently (e.g. as the results produced by the location detection module)
- As part of the disclosure scoring framework described in D6.4

The types of information that can be inferred by processing users' texts include a wide variety of personal information such as:

- Location trail, including home location and visited places that are estimated by using probabilistic location models from large quantities of geolocated training data.
- Favourite brands and products (e.g. mobile phones, clothes) that are mined through the identification of named entities (brand and product names) which appear in users' texts.
- User's stance on their areas of interest that is extracted using opinion mining tools which are adapted to a use with OSN multilingual and heterogeneous content.

A variety of personal information is shared on OSNs, including: (a) status updates added by the users, (b) comments on their multimedia content (i.e. photos, videos) and (c) third-party publicly available texts (i.e. newspaper articles, blogs etc.) that are shared by the users. Due to this variety of information, flexible and extensible text mining and linking modules and approaches need to be employed. The first stream of work is focused **on multilingual opinion mining** (Section 2) methods, which can be used to assign a polarity score (from very negative to neutral and very positive) to a text. Support for multilingual text processing is important given that a user may share content in different languages. A second significant approach is **entity linking** (Section 3), which deals with extracting named entity mentions from multilingual texts and linking them to entities from existing knowledge bases. Finally, improvements are brought to **location detection** (Section 4), which attempts to estimate the location(s) associated with a piece of text shared on OSNs.

Research on text mining and linking is part of the multidisciplinary research effort required by USEMP use cases and it is thus largely shaped by the conclusions of upstream research from other disciplines (notably legal studies, user studies and system design). In USEMP, more focus is put on visual content mining, which is more challenging, and a choice was made in the project's DoW to adapt a majority of text mining modules from existing

approaches that are aligned with the USEMP objectives and are also well mastered by project partners. As in the first iteration of the text mining tools, existing implementations of text mining were reused wherever possible. To assess the reliability and quality of the prototyped solutions, they were evaluated using suitable publicly available or generated datasets and in the case of entity linking and location detection, through participation in international benchmarking activities.

In D5.4, we continue to follow the guidelines for the implementation of technical components respected in D5.1 that stem from the use case analysis in D2.1 and the associated requirements defined in D2.2. In particular, the following requirements remain central:

- [SR02] “The system may be able to process the information within one second such that the user can make informed decisions on their past data without long delays. In the event data processing is to take longer, a progress bar should be presented. A maximal extent of 10 seconds will be aimed for.” This requirement has strong implications in terms of processing speed for the implemented components.
- [SR04] “The system may be able to make best effort associations between data placed onto OSN(s) and the profile attributes which can be inferred from such data.” This requirement is a counterpart of [SR02] that focuses on component performance, which should closely follow state of the art developments.
- [SR11] “The system may be able to get fruitful insights on how relevant a user’s profile is for different stakeholders.” Through inferences made by technical components, the end-users should be able to have insightful information on how her profile is seen by OSNs and, possibly, by other stakeholders.

A strong concern in USEMP is to provide users with a more complete view of how their data could be handled and exploited by OSNs. D9.3 showed that existing text mining tools are not tailored for privacy enhancement and, consequently, an adaptation step is needed in order to better satisfy domain requirements. Insights gained with D5.4 tools can be used both directly in the USEMP interface (D7.2), and as part of the privacy scoring framework described in D6.1 and D6.4, to complement social network mining inferences. For instance, a user’s view on a sensitive topic can be extracted from texts through opinion mining and can be displayed directly by the USEMP interface to inform the user about her degree of exposure on a certain privacy dimension (e.g. political beliefs).

2.Opinion mining

Opinion mining has rapidly become one of the most active topics in the field of natural language processing in recent years. This is easily explained by its commercial potential, notably in brand monitoring or gaining insight into the general population's view on public figures. In USEMP, we focus on noisy user generated content and go beyond the positive/negative binary labels. We predict for a text the probability of expressing a positive or negative opinion. This value can later be integrated into informative visualization tools, such as a colour gradient bar. We also approach the multilingual aspect of text processing here. Considering that in the first instalment of the USEMP text mining components (see D5.1), the focus was on English, we now propose a modular language independent framework for Web data opinion mining. Here, the single constraints are the availability of appropriate user generated training content and optional basic text pre-processing tools.

2.1. Related work

In the past years, automatic sentiment analysis of texts has attracted attention from both industry and academia. Such interest has produced a large body of research work, mainly focusing on the use of machine learning algorithms for opinion classification (Pang and Lee, 2008).

Most prior work with regard to opinion mining has been performed on standardized forms of text, such as consumer reviews or newswires. The most commonly used datasets include: news documents, web customer review data, Amazon review data or blogs. These corpora have also been identified as suitable for developing models on social media, in which we may encounter more informal text that poses additional challenges for Information Extraction and Natural Language Processing. Opinion mining on Social Media has recently started to receive a lot of attention from the scientific community. Several annotation projects have been proposed to support the development of sentiment analysis models for social media, focusing mainly on Twitter—one of the biggest initiatives being the SemEval 2013 task on the sentiment analysis (Nakov et al., 2013). Most of these datasets contain English documents only, while very few cover other languages. The majority of systems for sentiment analysis rely on the simple bag-of-words (BOW) representation. That is, the input text is split into n-grams of words). These are used in machine learning algorithms (e.g., Support Vector Machines (SVM) or logistic regression) to induce a model that can classify new instances. Such features can be effectively combined with external information, for example, with personalized co-occurrence statistics. While a few successful attempts have been made to use more evolved linguistic analysis for opinion mining, such as dependency trees and constituency trees with vectorised nodes, a study by (Wang and Manning, 2012) showed that a simple model using bigrams and SVMs performs on par with more complex algorithms.

Early opinion mining studies focused on the document polarity classification problem: for a given document, the algorithm assigns a label determining its general attitude (positive, negative or neutral). For some applications, this formulation may be simplistic and thus the most recent studies address more fine-grained tasks, including identifying subjective vs. objective parts of a document, opinion holders or more complex sentiments and emotions, in particular irony or sarcasm. Here, we work under the constraint of a multilingual setting,

making more complex approaches for opinion mining unapproachable. Thus, we focus on the first task proposed in this field, text polarity classification.

Modelling opinion mining as a supervised classification problem implies the need of abundant training data. A common approach consists of collecting a training dataset from certain websites. An ideal training sample should be representative in order to get good accuracy on heterogeneous data sources. These data, if not categorized properly, need to be manually labelled by human annotators, so that the opinions are associated with objective ratings. (Chaovalit and Zhou, 2005) found reviews from websites that provide ratings (e.g. Amazon) to be good candidates for training sets. Except for English, here labelled training data are freely available. For the rest of the targeted languages, we rely on the aforementioned methodology.

2.2. Method description

2.2.1. Overview

For the first iteration of the USEMP opinion mining module, we use a multilingual supervised approach, in which we automatically collect and clean training collections, if the latter were not available. We strive for a balanced performance between languages, thus we fix the same data source domain used for learning models: movie reviews. Opinion detection is applied on a short piece of text, that can range from a couple of words to multiple phrases and the predicted value is placed on a continuous scale from -1 to 1, with -1 representing *strongly negative*, 0 indicating a *neutral* text and 1 standing for *strongly positive*.

We propose an end-to-end framework for multilingual opinion mining that consists of two disjoint modules. The first one covers the offline stage of the framework and implements language agnostic learning models. The second module assures the online opinion score prediction step and first relies on an automatic language detector before selecting the use of the appropriate model learned in the previous stage. Since, we focus on four languages (i.e. English, French, Dutch and Swedish), a warning message is shown, if another language is detected. The training data acquisition step is decoupled from these two modules, as it depends on each data source. The languages implemented for this version of the framework were selected to cover the diversity of users that may take part in the living labs. However, new languages could be easily added, with the single constraint being the facility of obtaining training data.

Except English, for which we rely on publicly available datasets, we automatically collect training data from online movie reviews. In order to maximize the generalization potential of classifiers trained on movie reviews, we first clean the texts. To get to our final training collection, the following text cleaning measures are applied:

- **stemming**: We use the Snowball stemmer[1] for each language to remove the inflections and reduce the words to their root form.
- **proper noun removal**: As the text comes from the movie review domain, we remove the proper nouns (e.g. actors, movie titles, locations etc.) in order to reduce the

[1] http://snowballstem.org/

possible influence of such entities. Further attention was given to this issue by keeping at most 10 reviews per movie.

- **stop word/frequent words removal**: For stop word removal the common practice is the use of stop word lists. However, we go beyond stop words and remove very frequent words that do not bring discriminative information in the training process. The cut-off for what constitutes a frequent word is set at 90% (i.e. the top 10% most frequent words are discarded) after an initial round of classification experiments.
- **rare word removal**: Infrequent words expand the feature space, without improving the classification accuracy. The threshold for what constitutes an infrequent word is set at 10 (i.e. words that have less than 10 occurrences are discarded) after an initial round of classification experiments.

The next step is to train a classifier on the corpus. Once a supervised classification technique is selected, an important decision to make is feature selection. In text classification, features denote properties of textual data that are measured to classify the text, such as bag-of-words, n-grams (e.g. unigram, bi-grams, tri-grams), word position, header information, and ordered word lists. They can tell us how documents are represented. We experiment with three bag of words (BoW) language models in which terms are weighted by a *tf-idf* scheme: *unigram*, in which each word is treated independently, *bi-gram*, in which every sequence of two consecutive words found in the training corpus is taken into consideration and *unigram* + *bi-gram*, a combination of the two. Note that the broader *n-gram* representation can be instantiated with n = 3 or greater. While this has been shown to produce marginally better results for text classification tasks in experimental settings, in practice, an *n* larger than 2 is rarely used, due to very large feature spaces that slow the training and, most importantly, the prediction times.

2.2.2. Datasets

In USEMP, the textual user data that we analyse may come from one of the following languages: English, French, Dutch and Swedish. Considering that the availability of opinion mining labelled datasets for languages other than English is limited, we aim to extract new training collections for the remaining languages. In order to preserve comparability of our opinion mining module among languages, we use the same extraction and text processing pipeline and we target a single domain as a data source. As it has been shown in previous studies, movie reviews represent a valuable resource for deriving an opinion mining training collection (Chaovalit and Zhou, 2005). Having a single source domain also enables us to have a fair evaluation of the accuracy of each opinion classification model. We fix the English movie collection as a benchmark which we use the as a reference point when evaluation the usefulness of the collections gathered for other languages. Next, we detail the datasets that were used to train our opinion mining models for each language:

- **English**. We rely solely on publicly available collections. We first use the *MovieLens 1M Dataset*[2]. It contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000. The reviews have a score ranging from 1 to 5. We extract 50,000 reviews with the score 1 and label them as negative and 50,000 with the score 5, which are marked as positive. In order to

[2] http://grouplens.org/datasets/movielens/

add diversity to our data, and to be able to test the domain influence, we also use the *SNAP Review Dataset*[3]. It consists of reviews from Amazon and covers 25 product categories. The data span a period of 18 years, including ~35 million reviews up to March 2013. From this dataset, we use only 31,438 manually labelled reviews, following the same protocol described in (Liu et al., 2013).

- **French**. As there are no large available opinion mining datasets for French, we rely on an automatic extraction of movie reviews from the popular platform AlloCiné[4]. It uses a rating scheme between 1 and 5 stars, with a step of 1. We extract reviews with the rating 1, which are label as negative and reviews with rating 5, which are label as positive. This resulted in a collection of 100,000 reviews (50,000 positive and 50,000 negative).
- **Dutch**. We extract movie reviews from the MovieMeter[5] platform. It uses a rating format between 1 and 5, with a step of 0.5. We extract reviews with the rating 1 and 1.5, which are label as negative, and reviews with ratings 4.5 and 5, which are label as positive. We arrive at collection of 23,450 reviews (11,725 positive and 11,725 negative). While we are able to collect approximately 100,000 positive reviews, we chose to keep a balanced dataset by selecting as many positive reviews as available negative ones.
- **Swedish**. We extract movie reviews from the Nyheter24[6] platform. It uses a rating format between 1 and 5, with a step of 1. We extract reviews with the rating 1, which are label as negative and reviews with rating 5, which are label as positive. We arrive at collection of 55,200 reviews (27,600 positive and 27,600 negative). The size of the collection is limited by the availability of negative reviews. We collect the total number of negative reviews and we keep the same number of positive reviews.

In Figure 2.1, we show the distribution of randomly selected 1000 positive and negative reviews for the 4 languages. For English, the *MovieLens* corpus is used. The reviews are represented by a unigram vector model and the data are projected to a 2-D space using the t-SNE[7] algorithm. We can observe that the positive and negative reviews are globally well separated for English, French and Dutch. This preliminary observation is a strong cue that the automatically obtained data are coherent and there is no need for any manual intervention. This hypothesis is further consolidated by the experiments detailed in Section 2.3.

[3] http://www.text-analytics101.com/2011/07/user-review-datasets_20.html
[4] http://www.allocine.fr/
[5] http://www.moviemeter.nl/
[6] http://nyheter24.se/sok/filmset
[7] https://lvdmaaten.github.io/tsne/

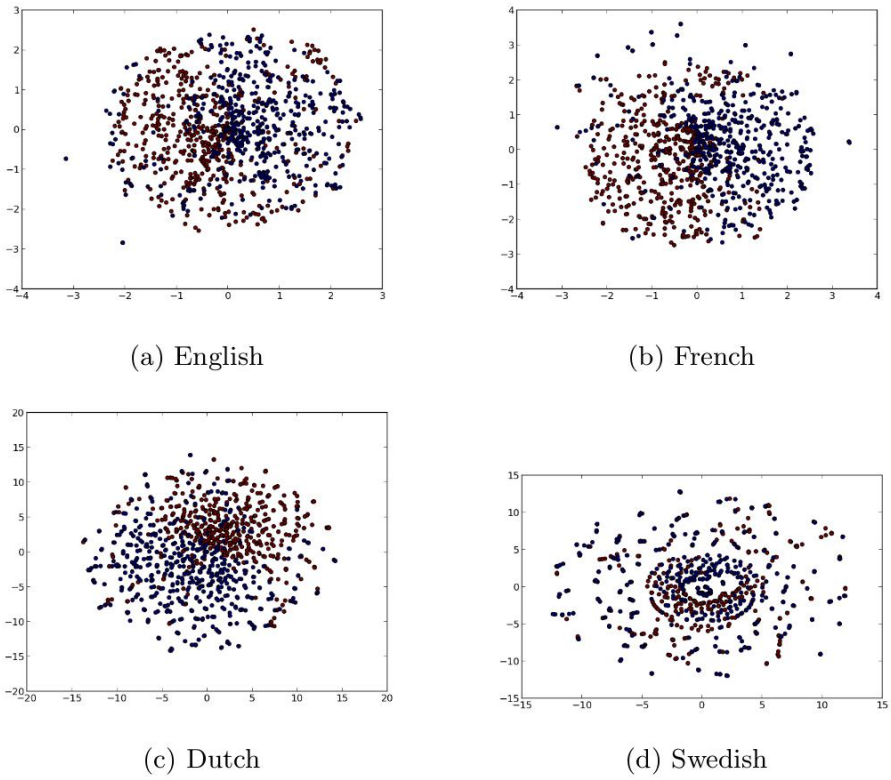

Figure 2.1: Distribution of 1000 reviews (500 positive and 500 negative) for English, French, Dutch and Swedish. The blue dots represent positive reviews, while the red ones indicate negative reviews. For English, reviews are taken from the MovieLens Corpus.

2.3. Evaluation

While the opinion mining module outputs a continuous score, we evaluate in a binary classification problem setting: predictions in the interval [-1, 0) are labelled negative and those that fall in the [0, 1) interval are considered positive. After an initial series of experiments, a linear support vector machines (SVM) classifier was chosen among others (e.g. Naïve Bayes, Random Forest, Gradient Boosting Classifier) to train the opinion mining models. Using linear classifiers is a common practice in text classification, where we deal with large dimensional spaces. Besides the competitive performance, the fast training and prediction times were also taken into consideration for practical issues. Parameter tuning is first performed on the MovieLens dataset and the best configuration is used on the rest of the collections. This collection was chosen as a reference as it is a publicly available reference dataset for English. The same classifier is used for the following experiments, in which we investigate the influence of the feature representation and the number of training instances, compare overall results between languages and datasets and look into the impact of training on data coming from a domain and testing on others. The experiments presented in this section should be seen as reference for the efficiency of our opinion mining framework but the final evaluation of its performance and utility will be given by the feedback provided by end users in the pilot studies.

2.3.1. Language model and training size influence

A major component in any text classification application is the choice of the feature representation. To facilitate the seamless multilingual approach of our framework, we disregard complex syntactic information and use solely frequency based language models. We compare three *tf-idf* BoW language models: *unigram, bigram and unigram+bigram*.

For the majority of the training collections, we are limited by the availability of positive or negative instances. Note that we strive for a balanced dataset for each language, so we reduce their size according to the minimum number of available examples between positive and negative samples. However, where available, we investigate the influence of the number of training instances over the accuracy on a test set. In order to assure a fair comparison of the training configurations (i.e. language model, number of training samples), besides the 100,000 reviews used for training we extract an extra 20,000 reviews (10,000 positive and 10,000 negative) from the *MovieLens* dataset for the test collection.

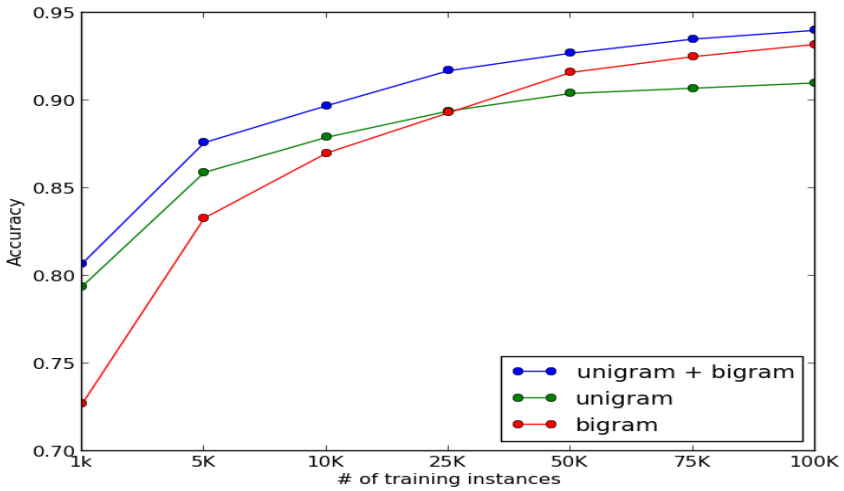

Figure 2.2: Influence of the number of training instances over the cross-validation accuracy scores for 3 language models on the MovieLens dataset.

From Figure 2.2, we can first observe that when combing unigram and bi-gram models to represent the texts, we get a higher accuracy on the test collection regardless of the number of training examples. However, the gain is marginally higher compared to the bigram representation when a large number of training samples are used (over 50,000). The unigram representation outperforms the bigram on in the setting where fewer training samples are used (less than 25,000). When looking over the impact of the number of training instances over accuracy scores, as expected, there is a constant increase of performance as more examples are used. While the unigram model reaches a plateau at 50,000 training samples, due to the larger diversity of features induced by the use of bigrams, a steady increase in accuracy is observed up to 100,00 instances. For the following experiments, we use the *unigram+bigram* model, while the number of training samples varies for each collection.

2.3.2. Overall results

In Table 2.1, we give an overview of the 5-fold cross validation accuracy results for each language. For English, we test on the *MovieLens* dataset (*EN/Movies* in the table) and the

SNAP Review dataset (*EN/Products* in the table), as well as on the combination of the two collections (*EN/Movies + Products* in the table).

	Language/Corpus					
	EN/Movies	*EN/Products*	*EN/Movies + Products*	*FR/Movies*	*NL/Movies*	*SW/Movies*
Size	100,000	31,438	131,438	100,000	23,450	55,200
CV Accuracy	**0.95**	**0.85**	**0.91**	**0.93**	**0.96**	**0.87**

Table 2.1: Training collection size and cross-validation accuracy results.

The distribution of examples among positive and negative reviews observed in Figure 2.1, where we can notice that for Swedish they are the least separable in a 2D space, is confirmed by the cross-validation experiments. Among the movie reviews datasets, for Swedish we obtain the lowest accuracy (0.87). Interestingly, although it has the fewest training examples, the highest average CV is reported for Dutch (0.96), suggesting the presence of highly discriminate words among the samples, while English, with a score of 0.95 and French, with 0.93 fall closely behind. Note that the English movies dataset is a publicly available collection that has been previously used in opinion mining research, while for the other three languages, we automatically collect and clean the data. The high and similar CV scores for all languages give us a first validation of our approach.

2.3.3. Domain shift influence

In USEMP, we process text content that is associated with OSN users, such as posts or comments. Except for English, to the best of our knowledge, there are no publicly available suitable labelled opinion mining datasets. In consequence, for the first iteration of this framework, we chose to use movie reviews as a multilingual Web data source. Although we clean the text used for model training and we disregard infrequent words for a better generalization, one drawback of our approach is the domain bias introduced by our choice of training data acquisition. However, this is a common problem in opinion mining and although it has been shown that domain adaptation techniques are beneficial, the gap between domain specific and models and generalized ones is not large in practice.

Product type	**Accuracy**	**Product type**	**Accuracy**
kitchen_&_housewares	0.817	baby	0.797
office_products	0.847	electronics	**0.778**
jewelry_&_watches	**0.861**	grocery	0.849
books	0.816	dvd	0.847
video	0.860	sports_&_outdoors	0.813
cell_phones_&_service	0.79	musical_instruments	0.83
camera_&_photo	0.793	health_&_personal_care	0.807
music	0.817	apparel	0.827
outdoor_living	0.813	magazines	0.812
gourmet_food	0.851	computer_&_video_games	0.795
beauty	0.83	toys_&_games	0.826
automotive	0.793	software	0.814
tools_&_hardware	0.815		

Table 2.2: Cross-validation accuracy results for multiple domains from the SNAP Review dataset.

We investigate the domain influence of our approach by training a model on the *MovieLens* dataset and testing the prediction accuracy for each product category from the *SNAP Review*

dataset. The results presented in Table 2.2 confirm the expected drop in performance when training and testing domains did not coincide. If we compare this with the best configuration illustrated in Figure 2.2, we notice a drop in accuracy that varies from 0.072 (prediction on the *jewelry_&_watches* category) to 0.155 (prediction on the *electronics* category). Despite this decrease, the actual prediction accuracy remains satisfactory, with only 6 out of 25 product categories scoring under 0.8. Also, the average accuracy for the *SNAP Review* dataset of the model trained on *MovieLens* is 0.82, compared to 0.85 cross validation test score on the full *SNAP Review* collection (see Table 2.1). This suggests that the lower scores are due to the intrinsic heterogeneous nature of the product reviews and the difficulty of predicting their polarity and less on the generalisation capacity of the model trained on movie reviews.

2.4. Implementation and usage

A Python implementation of the opinion mining module is provided. The scoring process can be launched by a single script call, as follows:

python score_opinion.py [models_folder] [text_file]

where *[models_folder]* represents the path to the folder containing the trained classification models for four languages: English, with the model trained on the combination of *MovieLens* and SNAP Review, French, Dutch and Swedish. The [*text_file]* parameter indicates the path to the texts for which the opinion is to be predicted. Each line of the input file has the following format*: text_id TAB text*. For each text, the language is first detected. If the identified language is not among the four previously mention, a warning message is raised. Otherwise, a score between -1 and 1 is assigned to each *text_id*.

In order to avoid loading learnt models in RAM for each call, we also provide a simple REST server implementation over the opinion mining framework using the *webpy*[8] module. The python REST server does not directly alter the Java backend of DataBait, and once the server is started the communication with DataBait is assured with localhost calls to the REST service. The python script is delivered with the required libraries and learnt models. In order to launch the service, the following command is run:

python run_opinion_service.py [port]

where *[port]* denotes the denotes the port on which the service should run. The service returns an opinion score for each text submitted through a call.

2.5. Next steps

We developed the first iteration of a multilingual opinion mining framework, in which we provide text level opinion prediction for English, French, Dutch and Swedish. Future improvements will be carried out in the following directions:

- Use the results from the entity extraction and linking framework and pass to fine-grained opinion mining for English. This entails identifying the target towards which the opinion is expressed.

[8] http://webpy.org/

- Enriching the training collections for French, Dutch and Swedish and extend them to more domains.

For the USEMP integration effort, the opinion mining framework will be added as a service in DataBait and will be evaluated during the user tests, as part of WP8 activities.

3.Entity linking

3.1. Introduction

We participated to the Entity Discovery and Linking track of TAC 2015. The goal of the Entity Discovery and Linking task in the TAC campaign is to extract named entity mentions from English, Spanish or Chinese texts and link them to entities existing in a knowledge base. For our first participation, we focused only on the linking part of the task (finding the correct entity in the knowledge base knowing the entity mention), for monolingual English text. In TAC EDL 2015, the reference knowledge base is a sample of Freebase, which introduces two new features. First, the developed system must deal with the challenge represented by the very large number of entities present in the knowledge base. Second, the entity linking systems usually exploit features associated with the entity that are either context-independent (such as string matching similarities) or context-dependent (Shen et al., 2015). The context-dependent features rely on a textual description of the entity (generally the content of its Wikipedia page). However, in Freebase, all entities do not come from Wikipedia and, therefore, are not always associated with a textual description. To tackle this problem (and generally try to improve the entity linking), we propose to add a context-dependent feature that takes into account the relation context of the entity in the knowledge base, i.e. the entities that are in relation with the candidate entity. We present in the following sections a more detailed description of our system and some evaluation results on both the DBpedia datasets used in previous TAC entity linking tracks and on the Freebase datasets of TAC 2015. We also discuss some error analysis we performed on these results.

3.2. Method description

3.2.1. Overview

In our participation, we want to test if using the relations between the entities in the knowledge base could improve the results of entity linking. We use a simple approach for entity linking, which performs the task independently for each query. The design of our system is quite standard (Ji et al., 2014): for each query, our system performs three steps: (1) analyze the query (entity mention and textual context) (2) generate candidate entities from the knowledge base (3) select the best entity among the candidates. These steps are presented more detailed in the following sections. We did not develop specific strategies for the last step required, to cluster the NIL entities: we used a simple clustering based on the string similarity of the entity mentions of the queries.

3.2.2. Knowledge base

In TAC EDL 2015, the knowledge base used is built from a Freebase snapshot. First, a filter is applied to exclude all the entities having one of the following types: book.written, work, book.book, music.release, music.album, tv.tv, series.episode, music, composition music.recording, film.film and fictional universe.fictional character: After filtering, around 8 million entities are left. We then imported the data (subject predicate object facts) into a relational database. The subject corresponds to an entity and is inserted in a table where each record is composed of the following attributes: the unique Wikipedia page title, the Wikipedia page id, the most notable type of the entity, the name in English and a tf-idf bag-

of-words vector representation of the Wikipedia page associated with the entity. The object can be either:

- **a literal**: the fact represents a property of the entity. It is stored in a table where each record is composed of three attributes: the subject identifier, the predicate type and the alphanumeric or numeric string attribute;
- **an entity**: the fact represents a binary relation between two entities. It is stored in a junction table where each record is composed the following attributes: the subject and object identifiers (entity identifiers in the entity table) and the predicate type;
- **a compound value type** (CVT): a CVT represents a n-ary relation which associates an entity with several other objects, that can be entities or literal attributes. They are stored in a CVT table, whose records are composed of the CVT identifier and its type. The relations are modelled by two tables: a junction table between the CVT and entity tables, composed of the following attributes: the CVT identifier, the predicate type and the object identifier (which is an entity identifier) and a CVT literal table, used to hold the relations where the objects are literal values.

Finally, the aliases and translations of every entity are inserted in a table where every record is composed of: entity identifier, alias or translation and language.

3.2.3. Query analysis

In the diagnostic task, each query is composed of an entity mention and the document in which this mention appears. We only considered the named entity mentions (NAM) and ignored the nominal mentions (NOM), since we did not include a co-reference resolution step for query analysis. In the query analysis step, we use the document to enrich the query, both for entity mention expansion and for context representation. More precisely, two kinds of expansion are performed, using named entities extracted from the document text by the MITIE9 tool:

- if the entity mention is an acronym, we search in the document named entities with matching initials and add them as variants of the entity mention;
- named entity mentions whose expression includes the target entity mention are added as variants of the entity mention.

For context representation, a tf-idf vector representation of the document is built, in the same vector space as the wikipedia documents from the knowledge base.

3.2.4. Candidate generation

Candidate entities are generated by comparing one of the forms of the query mention (either the direct entity mention or one of its variants found by acronym expansion or named entity similarity) and the KB entities using either (Dredze et al., 2010):

1. string equality with the normalized name of the entity in the knowledge base;
2. string equality with a variation (either an alias or a translation) of an entity in the knowledge base;

[9] https://github.com/mit-nlp/MITIE

3. approximate string matching with a variation of the entity in the knowledge base. In the submitted run, we use a simple string inclusion (the entity in the KB contains the targeted entity mention), since this functionality is directly available in the database;
4. approximate string matching (Levenshtein distance less than 2) with a variation of the entity in the knowledge base. For efficiency, we used a BKtree (Burkhard and Keller, 1973) for this functionality.

3.2.5. Candidate selection

Candidate features

For the selection of the best entity among candidates, we basically rely on two similarity scores. A first similarity score is based on the similarity between the textual context of the query mention and the textual context of the KB entity. More precisely, it is equal to the cosine similarity between the vectors representing the query document and the text associated with the candidate entity (i.e. its Wikipedia page). The second similarity score exploits the relations between the entities in the knowledge base. More precisely, we want to determine if the entities appearing in the text around the query mention are linked to the entities in relation with the candidate entity in the knowledge base. We adopted a simple approach to approximate this process (without having to perform the linking of the other entities in the document and to apply slot filling for verifying the actual presence of the relations in the document): for each entity E in the knowledge base, we build in the same vector space as the Wikipedia pages, a tf-idf vector containing the list of the entities in relation (either directly or through a CVT) with E. We then measure the relation similarity of the candidate entity by the cosine between this vector and the vector representing the query document.

Selection

For integrating all our criteria in a flexible way and choosing the best information to use for the selection of the best candidate, we relied on a statistical classifier. We added to the two similarity scores a set of four binary features that indicate the origin of the candidate generation (1 to 4 in section 4.2.4), with the idea that a candidate generated by direct string equality is stronger than a candidate generated by approximate string matching.

A classifier is then trained to recognize the best entity among the entity candidates, using the training data provided. More precisely, we used a binary classifier that decides, for each (query, candidate) pair if the query mention is an instance of the candidate entity. The positive examples are the instances taken from the training data, the negative examples are wrong candidates generated from the training data. Since the number of candidates generated for each query may be high (between 1 and 460,055), we limited the number of negative examples to be X times as big as the number of positive examples. In the submitted run, we used X = 10 and a Random Forest classifier. For every query, the classifier produces a probability for each candidate and the candidate with the highest probability can be selected.

Entity type filtering

The expected result of the EDL task must include the type of the entity, which must be one of the expected types: Person (PER), Geo-political Entity (GPE), Organization (ORG), Location (LOC), Facility (FAC). The query analysis step includes the use of the MITIE tool to extract

the named entity. But, the model we used only recognizes the types PER, LOC, ORG. Moreover, we did not want to solely rely on the quality of the named entity extractor for the entity type: we decided to generate the candidate entities without any constraint on their type. An additional filtering on the entity type is then added during the candidate selection process. More precisely, we keep the 5 best results returned by the classifier. Next, we filter out the candidate entities that do not have a type compatible with the expected types and then we select the remaining candidate with the highest score (if there are any). Simple ad-hoc rules have been used for the compatibility of the entity type found in the knowledge base and the expected types (e.g. 'administrative division' or 'country' are possible Freebase types for GPE). A query is finally marked as NIL if no candidate entity is found during the candidate generation step or if the classifier or the entity type filtering rejects all candidates.

3.3. Evaluation and testing

3.3.1. Test on DBPedia

Since this is the first time Freebase is used as a KB in the TAC Entity Linking task, we first developed our system using the DBpedia database that was used in previous years (2009 to 2013). Table 3.1 presents some statistics on the queries for these datasets.

	# queries	NIL queries	# candidates	NIL cand.	Avg. cand.	Cand. Recall
2009	3,904	2,229	208,060	949	70.41	84.0%
2010	2,250	2,230	232,672	601	141.10	89.4%
2011	2,250	1,126	329,508	388	176.96	87.9%
2012	2,226	1,049	420,179	117	199.23	92.4%
2013	2,190	1,007	394,217	395	219.62	83.5%

Table 3.1: Candidate statistics for the DBpedia datasets (TAC 2009 to 2013).

In particular, the candidates' recall, defined by the percentage of non-NIL queries for which the expected candidate is in the candidate list, seems quite good, for simple candidate generation strategies, and the number of candidates per query is also reasonable (the maximum number of candidates per query is between 2,718 and 9,964 depending on the years). Table 3.2 presents the results obtained by our system with different classifiers: Random Forest, linear SVM and Adaboost (we rely on the scikitlearn implementation of these classifiers, with no particular optimization of the parameters). We use as evaluation measures strong all match and recall (strong link match), that correspond respectively to the overall accuracy and the KB accuracy in the previous TAC EDL tracks.

	Strong all match			Recall (strong link match)		
	Adaboost	SVM linear	Random forest	Adaboost	SVM linear	Random forest
2009	77.4%	74.3%	70.8%	65.5%	65.9%	70.1%
2010	77.1%	80.4%	71.1%	73.6%	72.5%	70.4%
2011	71.9%	72.6%	61.1%	58.6%	58.9%	58.2%
2012	51.8%	50.4%	49.7%	46.6%	48.0%	47.3%
2013	73.8%	74.1%	68.8%	65.8%	67.1%	65.5%

Table 3.2: Entity Linking results on the DBPedia datasets, tested on one year and using all other years for training.

The results obtained with the proposed method are quite good, even if some datasets seem more difficult than others, such as the 2012 dataset (even if it is the year for which the candidate recall is the higher, the entities also seem to have more ambiguity – more candidates). The differences between the classifiers are not obvious: Adaboost and linear

SVM generally perform better on the overall measure, whereas Random Forest can be better on the strong link match measure (on the non-NIL entities).

3.3.2. Test on Freebase KB

In the TAC 2015 EDL track, both the KB and the number of queries (in the training data provided and in the test data) are much larger. Table 3.3 presents the candidate statistics on the training and test data. For the test data, we restricted the queries to English queries with named entity mentions (NAM). We also removed from these results the queries with entity type TTL that are ignored in the gold standard by the official evaluation program (these queries were in our submitted run). We can see in this table that the same candidate generation strategies generate many more candidates than in the DBPedia case (which can simply be explained by the size of the database), and the candidate recall is also lower.

	# queries	NIL queries	# candidates	NIL cand.	Avg. cand.	Cand. recall
Training	12,175	3,215	5,844,592	1,282	458.08	76.0 %
Test	13,587	3,379	6,141,369	1,255	480.32	77.6%

Table 3.3: Candidate statistics for the TAC 2015 EDL datasets.

We present in Table 3.4 our results, as computed by the official evaluation script. These results are different from the official results because they were computed on the gold standard restricted to the set of queries that we actually considered. The evaluation is then performed on the 13,587 remaining queries. Since we do not perform named entity recognition nor focus on the entity types, the results presented are restricted to strong nil match, strong link match and strong all match. We generally achieve a good score of almost 60% f-score results, but we can see that our system tends to produce too many NIL answers (precision on NIL answer is only 49%).

ptp	fp	rtp	fn	precision	recall	f-score	measure
5,464	2,810	5,464	4,744	0.660	0.535	0.591	strong link match
2,592	2,721	2,592	787	0.488	0.767	0.596	strong nil match
8,056	5,531	8,056	5,531	0.593	0.593	0.593	strong all match

Table 3.4: Results obtained on the EDL 2015 English queries

If we consider the f-score for the strong all match measure, we rank fourth of the campaign among 8 participants.

3.4. Next steps

We developed a full baseline system to address the problem of entity linking. The participation to the campaign TAC 2015 showed that our system has a satisfactory performance. The obtained results are already encouraging and future work is directed in two main directions, namely improvement of the quality of results and integration in USEMP. From a scientific point of view, we will focus on: (1) improving the current system by integrating state-of-the-art approaches to some of steps described above (2) improving the current system by developing innovative approaches; in particular, we intend to test a supervised classification approach to select the candidate (section 4.2.5) (3) going further by including a multimedia dimension to entity linking.

From a USEMP integration perspective, entity linking will be included in the architecture and used during the user tests as part of WP8 activities. Furthermore, the entity linking module included in the system will be updated when improvements are obtained as a result of scientific advances.

4.Location detection from texts

During the second iteration of development, a number of refinements were conducted on the location detection module, in particular regarding the selection of features (keywords) with improved geotagging performance, i.e. keywords that strongly correlate with geographic location. In addition, the first set of USEMP data, coming from the pre-pilot study were analyzed in order to identify performance issues (in terms of detection accuracy) of the module and to prepare an appropriate plan to address them.

4.1. Related work

Location detection from texts is a challenging task, which has attracted increasing research interest in recent years. Most efforts to date have focused on the textual information carried by social media content. However, these vary on the way that they model the problem and the source where they derive their data for training and testing.

A very popular method for location detection is the construction of geographical Language Models (LM) based on the textual metadata of user-generated geotagged items. The goal of such models is to link the presence of certain keywords to specific locations (typically cells on a rectangular grid) and make possible the estimation of geographic coordinates for new textual items. One of the earliest works (Serdyukov et al., 2009) used a predefined grid of cells and calculated the prior probabilities for image tags based on the neighbourhood of the cells where they appeared. More recent research by (Van Laere et al., 2013) explored various approaches based on different clustering, feature selection schemes and language models. Furthermore, in (Van Laere et al., 2014) they proposed improved term selection techniques, utilizing kernel density estimation and Ripley's K statistic, to enhance the accuracy of geotagging. Another recent effort by (Liu et al., 2014) is based on the original approach by (Serdyukov et al., 2009), and introduces a user profile framework to combine the image tags with their users. User profile is represented by the set of historical tags and it is linked to a distinct geographic location using a similarity measure.

Another way to tackle the problem is through the discovery of geographical topics in a corpus of multimedia items that contain textual information. One of the early works following this scheme was presented by (Eisenstein et al., 2010) and proposed the use of a multi-level generative model for jointly mining latent topics and geographical regions. In (Yin et al., 2011), a geographical topic discovery approach was proposed using a Latent Geographical Topic Analysis (LGTA) model comprising location and textual information learned from a dataset of geotagged Flickr images. Additionally, in (Hong et al., 2012) an algorithm was developed that used the Twitter stream to model the geographic location of tweets based on topical, geographical, and interest distribution of users.

4.2. Method description

The objective of text-based location detection is to estimate the geographic location of a post using text analysis on its content. To this end, we build a Language Model (LM) using a massive amount of geotagged items as a training set. Additionally, compared to the first version of the module (described in D5.1), we developed a framework for feature selection and weighting to increase the robustness of the model and reduce its size. To ensure more reliable estimations, we employ a similarity search method and a multiple resolution grid

technique. More details regarding the approach and the conducted evaluation are included in (Kordopatis-Zilos et al., 2015) and in an upcoming journal article submission.

As has already been described, the LM is built on a grid of cells (i.e. nearly rectangular) of size 0.01° of latitude and longitude corresponding to approximately 1km^2, near the equator, based on the scheme of (Popescu, 2013). The geotagged items used for creating this model are Flickr images from the CC-licensed Yahoo Flickr 100M dataset. In particular, the tags of the geotagged Flickr images are used to build a term-cell probability structure based on the user count of each term in the individual cells. The cell probability, for a query item (i.e. the post to be geolocated) that carries an arbitrary number of terms is derived from the summation of the term-cell probability in every individual cell. The cell with the greatest probability is considered to be the **most likely cell** (mlc) for the query item and is used as the base location estimate of the model.

Focusing on more accurate prediction in finer granularities, we also built an additional language model using a finer grid (cell side length of 0.001° corresponding to approximately 100m) and applied an Internal Grid technique to fuse the estimations of the two models, selecting for a query item the finer grid cell if considered reliable, otherwise the coarser grid cell. Then, employing the Similarity Search technique from (Van Laere et al., 2011), we compute the textual similarity between the query item with every item of the training set inside the mlc, based on the Jaccard similarity of the corresponding sets of terms. The final location is derived by computing the center-of-gravity of the k most textually similar posts to the query post.

4.2.1. Feature Selection

Due to the massive size of the LM as well as the significant amount of terms lacking geographical interest, a feature selection scheme is necessary. To this end, the terms are sorted and filtered based on certain criteria. The main criteria pertain to the geotagging capabilities of the terms and their spatial-awareness.

To evaluate the geotagging capability of a term, we apply a technique inspired by cross-validation (using the items composing the training set only). First, we partition the training data into p folds. Subsequently, one partition at a time is withheld, and the rest $p-1$ partitions are used to build the language model and calculate the prior estimations for the items in the withheld partition. Thus, the accuracy a of every term is the ratio of the number of correctly geotagged items in range *r* where the term appears over the total term occurrences and is considered as an indicator for every term's geotagging strength. The terms with non-zero accuracy score are forming a set denoted as T_a.

Locality is the metric we devised to quantify the spatial-awareness of terms based on the different users that used the same term across the grid. For every individual term, the locality score is calculated based on the term user count and the neighbour users that have used it in a geographically distinct area. More specifically, every time a user uses a specific term, he/she is assigned to the respective grid cell. As a result, a set of users is formed for each cell and they are considered neighbours (for that particular cell). Then, the total number of neighbours of every user on every cell are summed up and divided by the square of the total user count, which is considered as the neighbour probability of the term. Finally, locality derives from the multiplication of the neighbour probability with the total user count. Locality is computed as:

$$l(t) = N_t * \frac{\sum_{c \in C} \sum_{u \in U_{t,c}} \left| \{u' | u' \in U_{t,c}, u' \neq u\} \right|}{N_t^2}$$

where $l(t)$ is the locality score of term t, N_t the total user count of t, C denotes the set of all cells and $U_{t,c}$is the set of all users that used term t inside cell c. Eventually, the final set T used by the approach is the intersection of the two sets: $T = T_{a \cap l}$.

4.2.2. Feature Weighting

In order to adjust the original LM term probabilities for each cell, we weight the terms in T based on their locality and spatial entropy. Having computed the locality scores, we sort the terms based on their scores and calculate their weights using their position in the distribution. The locality weights are computed by the following equation:

$$w_l = \frac{|T| - (j-1)}{|T|}$$

where w_l is the weight value of term t on the j[th] position in the distribution and $|T|$ is the total number of terms contained in T.

Additionally, for each term in the model, its spatial entropy value is calculated based on the Shannon entropy formula applied on their term-cell probabilities. After the calculation, a Gaussian weight function is generated obtaining the mean value and standard deviation of the spatial entropies distribution. Finally, the spatial entropy weights are normalized and defined as w_{se}. To combine the two weights, we apply the following weighting scheme:

$$w = \omega * w_{se} + (1 - \omega) * w_l$$

4.3. Evaluation and testing

4.3.1. MediaEval 2015 Placing Task

The approach was evaluated as part of the MediaEval Placing Task 2015 challenge[10], which is an annual international benchmarking initiative and its main objective is dedicated to the geo-localization of multimedia items using a corpus of geotagged data. The data used in the challenge comprised images and videos from the released YFCC dataset[11]. Participants were challenged to estimate locations (in terms of latitude and longitude) of the 949,889 geotagged items that are contained in a test set using another set for training of ~4.7 million items, both sets released by the organizers. Moreover, participants were asked to submit up to five runs, including at least one text, one visual and one hybrid (using both text and visual features). The evaluation of the runs was based on their precision in range *r* (percentage of correctly placed items within range *r*) and their median geotagging error (median of the error distribution).

In this section, the two submitted text runs are described since they fall within the scope of this report. To obtain the estimates for these runs, we built the LM using the scheme of section 5.2.1, a probabilistic LM was built on a grid of rectangular cells of size 0.01°, applying feature selection and feature weighting, with $p = 10$ and $\omega = 0.2$, respectively, both selected

[10] http://www.multimediaeval.org/mediaeval2015/placing2015/
[11] http://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67

empirically on the training set. To ensure more reliable estimations in finer granularities, we employed the Internal Grid technique and Similarity Search using $k = 5$.

For comparison reasons, we present the following runs:

- RUN_1 - probabilistic LM with all refinements using the dataset of 4.7M items released by the organizers for training.
- RUN_4 - probabilistic LM with all refinements using the full YFCC dataset (but excluding all items form the users that appeared in test set).
- Simple run - base LM without any refinement using the released dataset.
- State-of-the-art - organizers' baseline run based on (Van Laere et al., 2013).

We present the results corresponding to these runs in Table 4.1. The best results are obtained by the probabilistic LM with all refinements when applied on the extended training set (RUN_4). This confirms the hypothesis that the use of more data for training has beneficial impact on the performance of the approach, since the precision in all ranges is considerably increased and the median error dropped more than 65% in comparison with RUN_1. Additionally, the application of all the refinements leads to significantly better results in any aspect. Finally, the approach is highly competitive since it outperforms the state-of-the-art (organizers' run) in four out of five precision ranges and has slightly lower median error.

Run	P@10m	P@100m	P@1km	P@10km	P@100km	m. error
RUN_1	0.61	6.40	24.33	43.07	51.08	69
RUN_4	**0.75**	**7.73**	**27.30**	**46.48**	**54.02**	**24**
Simple run	0.02	0.64	21.78	37.68	44.41	342
SoA	0.49	4.23	18.44	39.96	51.33	71

Table 4.1. Geotagging precision (%) for five ranges and median geotagging error (km) for the two submitted text runs (RUN1, RUN4), one run based on only the probabilistic LM using the released dataset (Simple run) and the baseline of the task (SoA – State of the Art) submitted by the organizers.

4.3.2. Pre-pilot Data

Additionally, we applied the method on the Facebook data collected from the pre-pilot study conducted with the help of DataBait. The algorithm was fed with one post at a time and it estimated the most probable location for this post. The estimations were carried out using the prior probabilities of the LM that was built from the entire YFCC, having applied the feature selection and feature weighting schemes described above.

However, since there is no ground truth for the posts in this dataset, we performed manual evaluation: every post that contained a toponym, a name of a landmark or a geographical term was regarded as *geolocated*. For every such post, we used the earth surface area corresponding to the specific geographic term as ground truth of the model. In particular, we considered two discrete groups of geolocated posts based on the spatial awareness of their terms. The first group comprises posts, of which the location is derived from toponym terms such as the names of countries or big cities. Such posts are referred to as *limited-geolocated*. The second group contains posts comprising posts, of which the location is determined by terms corresponding to small cities (with a surface area of less than $15km^2$), landmarks (i.e. monuments, stadiums, etc.), or geographical landscapes (e.g. lakes, rivers, etc.). Those are referred to as *well-geolocated*. Finally, if more than one geographic term appeared in a post, then we associated the post with the geographic area of the most spatially-aware term. For example, for a post that contains a country, a city and a

neighbourhood name, we annotated it with the location of the referred neighbourhood (which is more specific than the location of the mentioned city and of course country).

For the evaluation of the model, we treat the aforementioned two groups of posts differently. For the limited-geolocated posts, the estimated coordinates have to fall inside the borders of the respective area in order for them to be considered correctly placed. In contrast, the estimations of the well-geolocated posts have to be within a range of 1.5km from the ground truth to be considered correctly placed.

Since manual evaluation is a very time consuming process, we manually evaluated only a small random fraction of the entire dataset. Table 4.2 illustrates the results of approach. The total amount of the evaluated posts was 5404. From this set, only 646 posts, i.e. 11.96% of the sample, was found to be *geolocated*, either well-geolocated (3.07%) or limited-geolocated (8.89%). From those posts that were actually *geolocated*, 70.9% were placed correctly, which implies that the algorithm works reasonably well.

More precisely, 70.63% of the limited-geolocated posts were correctly placed. The major reason of misplacing this type of posts was due to the extensive number of terms. Typically, the probability of the geographical terms of such posts is distributed among a wide area. Consequently, when they occur with multiple non-geographical terms, their impact on the estimation process diminishes. Moreover, the method generated slightly more accurate estimated locations for the well-geolocated posts, reaching a precision of 71.69%. In this case, the main factor that led to incorrect estimation was the lack of corresponding geographical terms in the training used to build the LM. For instance, the names of small unpopular countryside areas (e.g. the Norrtälje municipality near Stockholm, the Humelgem area near Brussels airport, and the Strombeek-Bever Belgian town) were not included in the LM that was constructed from the YFCC dataset.

Group	Precision	Fraction
limited-geolocated	70.63	3.07
well-geolocated	71.69	8.89
overall	70.90	11.96

Table 4. 2 Geotagging precision (%) and the fraction (%) with respect to the annotated set of 5404 posts to which each subset corresponds, namely the two geolocated post groups (limited-geolocated, well-geolocated), and the total number of the geolocated posts.

As was mentioned above, the total amount of annotated posts was 5404, with 646 actually corresponding to some location. Yet, the initial version of the module provided estimations for 4988 of the posts, i.e. for many posts even when no location was associated with them. To alleviate this issue, we developed a simple thresholding scheme for avoiding producing a location estimate for such a high number of posts. For this thresholding, we used the maximum locality and minimum spatial entropy of the terms of a post. To assess the effectiveness of this scheme, we carried out experiments studying the impact of either locality or spatial entropy thresholds on the location detection performance. Note that for locality, a post was considered geolocated when the maximum locality of its terms exceeded the threshold, while for entropy a post was considered geolocated when the minimum spatial entropy of its terms was below the threshold. We defined two evaluation measures: the overall precision, which is the percentage of correctly placed posts in the selected set (i.e. number of correctly placed posts over total number of predicted locations), and the recall, which is the percentage of the correctly placed posts (out of the set of 646 posts that were associated with a geographic location). The obtained performance is illustrated in Figure 4.1.

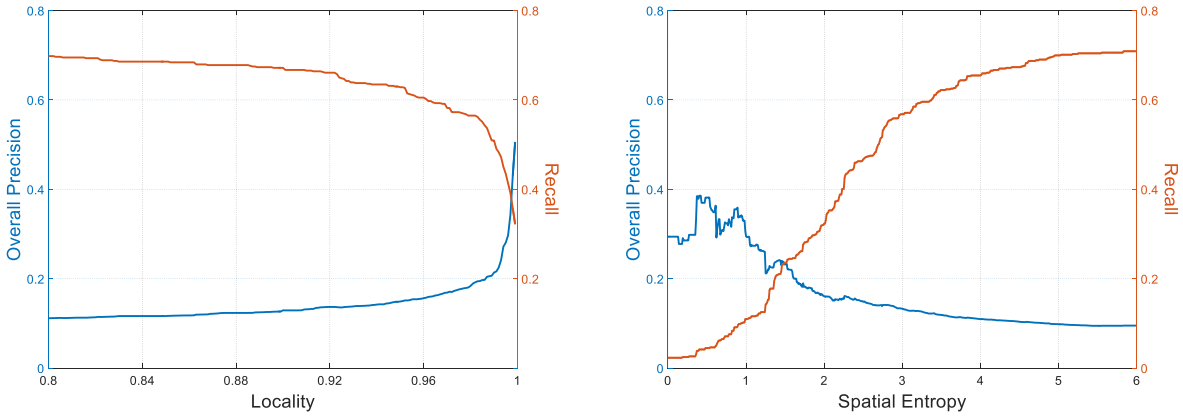

Figure 4.1 Overall precision and recall of the post selection experiments relative to the threshold value of post's locality and spatial entropy.

For locality, setting a threshold lower than 0.8 did not affect performance; hence, we are presenting the result of the experiments with thresholds greater than this value. As the selection procedure became stricter, overall precision steadily increased with proportional reduction of recall. In threshold values greater than 0.99, overall precision reached up to 50%, with a significant decrease of recall to approximately 32%. On the other hand, performing the filtering of posts using their spatial entropy as indicator was found to be suboptimal. For thresholds greater than 2.7, overall precision firmly dropped while recall steadily increased. However, in low spatial entropy values, overall precision fluctuated between 25-40%, while recall decreased at a massive rate.

Finally, it is noteworthy that in a lot of cases the user's home location, as stated in the user's questionnaire, was correctly estimated repeatedly in their posts. This observation gives the opportunity to explore a similar problem which is related with the geolocation of the home location of users.

4.4. Implementation and usage

Similar to the previous edition, a Java implementation of the probabilistic LM approach of section 5.2 was delivered. The implementation is based on the LM generation process applying the feature selection and feature weighting schemes described in 5.2. The Internal Grid technique and Similarity search are not included, since they are computationally expensive and more than double the size of the model.

For convenience, the provided implementation is distributed in the same format as in the previous deliverable (D5.1). Thus, together with the implementation, we make available a pre-computed probabilistic LM. Since the model was generated using the entire YFCC dataset, several filtering operations were applied alongside the feature selection in order to make the model more compact (i.e. stop-words filtering, removal of terms that contain the symbol "+" etc.). Still, the model would require approximately 4GB of main memory to be fully memory-based. To generate location predictions for a set of input text messages, the following command should be executed (assuming a JRE is installed):

java -Xms4G -Xmx4G -jar geopred.jar [root-folder] [text-input-path] [output-path]

where [root-folder] denotes the folder where the library and location model files reside, [text-input-path] is the path to a text file containing the input texts (one per line), and [output-path] is the path to a text file containing the predicted locations (one per line), which will contain

the estimated location (in terms of latitude, longitude), the closer city and country in that location and the evidence terms that led to this estimate (comma-separated). The form is:

latitude \t longitude \t city \t country \t evidence_(comma-separated)

In the initial form of the model, executing the above command would load the LM in memory and then perform the prediction. For sets of input texts that are small or moderate in size (e.g. a few hundreds to thousands), this would clearly result in unacceptable overhead (since loading the full LM in memory typically takes between two and three minutes). Hence, for implementation within the DataBait backend, the reverse geocoder is now loaded into memory on start-up in a singleton bean which server components can access fast during operation. In particular, the LM is mapped to an in-file system with an object cache to speed up common queries. In addition to working with partial memory structures, this significantly reduces the memory footprint (to under 100 MB with dynamic cache), this increases the speed for regular word look-ups, and speeds up server turn around and replication time, as the significant data structures required for the LM do not have to be loaded into memory.

4.5. Next steps

The results of the experiment carried out from the pre-pilot dataset appear promising, yet much of the user feedback pertaining to location estimates was critical. Hence, future work will focus on further improving location accuracy and in addition exploring the problem of detection of user's home location or even the locations that a user has visited recently.

5.Conclusions and future work

During the second iteration of the project, work on developing textual mining and linking modules was improved and extended in three main directions: a) a supervised end-to-end framework for multilingual opinion mining was implemented and opinion prediction models were trained for English, French, Dutch and Swedish b) a simple approach for entity linking was proposed, in which the relations between the entities in the knowledge base are explored independently on each query c) several improvements were performed on the location detection module, in particular regarding the selection of keywords with improved geotagging performance.

While for the opinion model framework we carried out an internal evaluation through cross validation on datasets that are publicly available or that were automatically collected, the entity linking module was evaluated in the Entity Discovery and Linking track of the TAC 2015 evaluation campaign, where we obtained competitive results. For location recognition, we introduced a simple and scalable formulation of probabilistic location models and combined them with other cues. Experimental validation done as part of the MediaEval Placing Task 2015 showed that our method clearly outperforms the state-of-the art results provided by the organizers. This approach was also manually tested on the Facebook data collected from the pre-pilot study conducted with the help of DataBait.

From a scientific perspective, several improvements are envisioned on the backbone of the existing text mining modules: a) enriching the training collections for French, Dutch and Swedish followed by their extension to more domains and implementing fine-grained opinion mining methods for English b) improving the current entity linking system through the use of a supervised classification approach to select the candidate and going further by including a multimedia dimension to entity linking and c) further improving location accuracy and in addition exploring the problem of detection of user's visited or familiar locations. In parallel to improving and extending the text mining modules, we will focus on their integration in the USEMP system. We will progressively integrate the other modules with the overall objective of reaching full integration by the end of the final reporting (September 2016).

6.References

P. Chaovalit, L. Zhou. (2005). Movie review mining: A comparison between supervised and unsupervised classification approaches. In System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on (pp. 112c-112c). IEEE.

M. Dredze, P. McNamee, D. Rao, A. Gerber, T. Finin. (2010). Entity disambiguation for knowledge base population. Proceedings of the 23rd International Conference on Computational Linguistics.

J. Eisenstein, B. O'Connor, N. A. Smith, E. P. Xing. (2010). A latent variable model for geographic lexical variation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (pp. 1277-1287).

L. Hong, A. Ahmed, S, Gurumurthy, A. J. Smola, K. Tsioutsiouliklis. (2012) Discovering geographical topics in the twitter stream. In Proceedings of the 21st international conference on World Wide Web (pp. 769-778). ACM.

H. Ji, J. Nothman, B. Hachey. (2014). Overview of tac-kbp2014 entity discovery and linking tasks. In Proc. Text Analysis Conference (TAC2014).

G. Kordopatis-Zilos, A. Popescu, S. PapadopoulosY. Kompatsiaris. (2015). CERTH/CEA LIST at MediaEval Placing Task 2015. Working Notes Proceedings of the MediaEval 2015 Workshop, Wurzen, Germany.

B. Liu, Q. Yuan, G. Cong, D. Xu. (2014). Where your photo is taken: Geolocation prediction for social images. Journal of the Association for Information Science and Technology, 65(6), pp. 1232-1243.

B. Liu, E. Blasch, Y. Chen, D. Shen, G. Chen. (2013). Scalable sentiment classification for big data analysis using Naive Bayes Classifier. In Big Data, 2013 IEEE International Conference on (pp. 99-104). IEEE.

P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, T. Wilson. (2013) . SemEval-2013 Task 2: Sentiment Analysis in Twitter, in: SemEval.

B. Pang, L. Lee. (2008) . Opinion mining and sentiment analysis, Foundations and trends in information retrieval 2 (1-2) 1–135.

A. Popescu. (2013). CEA LIST's participation at MediaEval 2013 Placing Task. In Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, Barcelona, Spain.

P. Serdyukov, V. Murdock, R. Van Zwol. (2009) . Placing Flickr photos on a map. Proc. Of ACM SIGIR 2009.

W. Shen, J. Wang, J.Han. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. Knowledge and Data Engineering, IEEE Transactions on, 27(2), pp.443-460.

O. Van Laere, S. Schockaert, B. Dhoedt. (2013) . Georeferencing Flickr resources based on textual meta-data. Information Sciences, 238, 52-74.

O. Van Laere, J. Quinn, S. Schockaert, B. Dhoedt. (2014). Spatially aware term selection for geotagging. Knowledge and Data Engineering, IEEE Transactions on, 26(1), 221-234.

Z. Yin, L. Cao, J. Han, C. Zhai, T. Huang. (2011). Geographical topic discovery and comparison. In Proceedings of the 20th international conference on World Wide Web, pp. 247-256.

S. Wang, C. Manning. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification, proceedings of ACL (Association for Computational Linguistics).