### Deliverable D4.1

<table>
<thead>
<tr>
<th>Work-package</th>
<th>WP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deliverable</td>
<td>D4.1 First Release of Text-analysis Technologies and Models</td>
</tr>
<tr>
<td>Deliverable Leader</td>
<td>SAIL</td>
</tr>
<tr>
<td>Quality Assessor</td>
<td>CERTH</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>Public</td>
</tr>
<tr>
<td>Delivery date in Annex I</td>
<td>M12, May 31, 2020</td>
</tr>
<tr>
<td>Actual delivery date</td>
<td>May 29, 2020</td>
</tr>
<tr>
<td>Revisions</td>
<td>2</td>
</tr>
<tr>
<td>Status</td>
<td>Final</td>
</tr>
<tr>
<td>Keywords:</td>
<td>Text Analytics, Tokenization, Information Enrichment, Sentiment Analysis, Named Entity Recognition, Topic Clustering, Detection of Computer-generated text</td>
</tr>
</tbody>
</table>
Disclaimer

This document contains material, which is under copyright of individual or several MIRROR consortium parties, and no copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the MIRROR consortium as a whole, nor individual parties of the MIRROR consortium warrant that the information contained in this document is suitable for use, nor that the use of the information is free from risk, and accepts no liability for loss or damage suffered by any person using this information.

This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein.

© 2020 Participants in the MIRROR Project
List of Authors

<table>
<thead>
<tr>
<th>Partner Acronym</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAIL</td>
<td>Gerhard Backfried, Erinç Dikici, Christian Schmidt</td>
</tr>
<tr>
<td>LUH</td>
<td>Erick Elejalde</td>
</tr>
<tr>
<td>FOI</td>
<td>Joel Brynielsson, Fredrik Johansson</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction .......................................................................................................................... 7

1.1 Purpose and Structure of the Deliverable ......................................................................... 8

1.2 General Data Protection Considerations .......................................................................... 8

1.3 Relationship to other Deliverables and Positioning of WP4 ........................................... 8

2 Language Identification ........................................................................................................ 10

2.1 State of the Art ................................................................................................................ 10

2.2 Relevance to and Approach within MIRROR .................................................................. 12

2.3 Component Integration ..................................................................................................... 13

2.4 Models/Technologies/ Insights available/under Development for MIRROR .................... 13

3 Extraction / Cleaning / Tokenization ..................................................................................... 13

3.1 State of the Art ................................................................................................................ 13

3.2 Relevance to and approach within MIRROR .................................................................. 14

3.3 Models/Technologies/Results/Insights/Next Steps .......................................................... 16

4 Named Entity Detection / Concept Detection ..................................................................... 16

4.1 State of the Art ................................................................................................................ 16

4.2 Relevance to and Approach within MIRROR .................................................................. 22

4.3 Component Integration ..................................................................................................... 22

4.4 NER Models and Technologies available/under Development for MIRROR .................... 23

4.5 Next Steps ....................................................................................................................... 24

5 Sentiment Analysis .............................................................................................................. 24

5.1 State of the Art ................................................................................................................ 24

5.2 Relevance to and approach within MIRROR .................................................................. 27

5.3 Component Integration ..................................................................................................... 28

5.4 Models/Technologies/Results/Insights/Next Steps .......................................................... 28

6 Document Clustering .......................................................................................................... 29

6.1 Related Work .................................................................................................................. 29

6.2 Unsupervised Neural Topic Modeling ............................................................................. 30

6.3 Supervised Deep Topic Modeling with Personalized Attention for News Articles ............ 32
7  Detection of Computer-generated Content ................................................................. 33
    7.1  Automatic Text Generation .................................................................................. 33
    7.2  Detection of Computer-generated Text ............................................................... 35
    7.3  Component Integration ....................................................................................... 36
    7.4  Ongoing Technical Work .................................................................................... 36
8  Summary and Next Steps ......................................................................................... 36
9  Bibliography ............................................................................................................... 38
Executive summary

This deliverable describes the first version of text-processing technologies comprising the extraction and pre-processing from source-platforms, sentiment-analysis, the detection of named-entities, topic-detection, document-clustering and the detection of computer-generated content. It provides the background, context and state-of-work for all technologies and components involved in the processing of textual contents within MIRROR.

Section 1 provides an introduction to the objectives of the work-package and how it is positioned with regard to the overall project and MIRROR architecture.

Sections 2-7 provide detailed descriptions of the individual technologies and components (one component implementing a technology) being developed within WP4. Overall, these components also correspond to the individual tasks of WP4. Each component is described concerning the current state-of-the-art, particular and specific challenges and settings within MIRROR and the domain of migration as well as providing initial results and insights gained from work of project year one where possible. All components will eventually be dockerized for actual deployment within the overall MIRROR system.

Chapter 8 concludes by summarizing the progress made so far and outlining the plans for future work.
1 Introduction

WP4 is devoted to the development and adaptation of a set of text-analysis technologies for the enrichment of information collected from traditional and social media (DoW). As such work comprises the integration of existing technologies and adaptation of these to the specific domain of migration as well as the development of genuinely new approaches for the various sub-tasks targeted by the WP.

Even though the amount of multimedia content has been growing dramatically over the past decades (fueled by the introduction of Social Media (SM) platforms, mobile devices and an always-on mentality of users), textual contents are still omnipresent and still comprise a large share of all content produced.

With the advent of SM, users effectively turned into prosumers, not only consuming content but also producing it continuously. As a result, the kinds and styles of content found on such platforms varies dramatically from that commonly found on traditional platforms.

The task of processing (enriching) data from both kinds of platforms, SM as well as traditional media, consequently poses additional challenges to any technology involved with regard to robustness and scope. Specific processing and extensions may be required to handle the often spontaneous and informal nature of SM.

The domain of migration and particularly the aim to detect and classify mis-perceptions about Europe in the eyes of non-Europeans requires not only to take into account this variety of platforms, but also requires technologies which are able to deal with the large diversity of languages (and dialects) encountered in this domain by persons and organizations communicating about migration-related issues. Specific terminology and phraseology are employed which pose further challenges to any automatic processing schemes addressing these topics.

The components developed and extended within WP4 all aim to address the above issues to the extent possible within the project.

Technologies and components dealing with the extraction and collection of data, allowing to access essential sources and language identification (LID) – to be able to attribute a language to a given document for appropriate further processing – form the basis of further enrichment. Enrichment-components such as Named Entity Recognition (NER), Sentiment Analysis (SA), Document Clustering (DC) and components aiming to determine the level of authenticity build upon these to yield higher-level information on aggregated levels.

This enriched content feeds into WP9 and the detection of possible mis-perceptions and threats. The components themselves form part of the overall MIRROR architecture; their technical integration into the architecture and workflows are discussed in WP7.

Work in WP4 spans a wide variety of technologies and considers many approaches. Some of these are in a rather mature state and are getting close to becoming running (docker) containers. For them, incremental work evaluating performance and improving it provides the road ahead. Other technologies and approaches have focused on more fundamental questions in project year one. As a consequence, results have only begun to emerge and work has been focusing more on laying the foundations. The flexible and agile approach chosen for the overall MIRROR system allows for phasing-in of components, staging them over time as they become available and improving them as progress is made.
1.1 Purpose and Structure of the Deliverable

The purpose of this deliverable is to provide the background and overview of the respective technologies and components which are being developed and/or extended with work performed in WP4. The respective state-of-the-art, approaches chosen and the relevance to the MIRROR project is emphasized throughout the document. The work which has been carried in project year one is presented. First results are provided where possible. Further evaluations and insights will be provided by the follow-on deliverables.

Chapters 2-7 describe the state-of-the-art, challenges and methods investigated and developed (or envisioned to be developed) within the individual tasks of WP4. Chapter 8 concludes with a summary and provides an outlook of activities to come.

1.2 General Data Protection Considerations

The methods and technologies developed in WP4 and described in this document process only textual data which enters the MIRROR system via the Data Manager container. The Data Manager container follows the "privacy by design" principle, separating the storage of personal information and the storage of data to be analyzed with the purpose of limiting access to personal data only to authorized subjects. The output of the analyses performed by the Text Analysis component does not in any way extract or identify personal information from the textual data. Any names or other personal data marked by any components are substituted by fictitious ones by the Pseudonymiser component, so no real names are exposed. Any data imported from the Media Mining System originates in open and publicly accessible sources only. Furthermore, it will have undergone adequate processing conforming to the principles as laid out in article 5 of the GDPR. More details about the data protection considerations of the Data Manager and the Pseudonymiser component can be found in Deliverable D7.1.

1.3 Relationship to other Deliverables and Positioning of WP4

Whereas this deliverable describes the text-based technologies being developed and/or extended within MIRROR, D5.1 provides an overview from the point of view of the Audio/visual Analysis (comprising the respective components). D7.1 provides information about the architecture and workflow which the components described within the present deliverable will run. The actual integration of components will be performed according to the specifications developed within WP7.

The overall architecture of the MIRROR systems is depicted in Figure 1 below.
For orientation, Figure 2 below (from the DoW) depicts the position of WP4 within the MIRROR project: its goals are determined to a large extent by the demand analysis carried out by WP2 and the feedback and insights generated by WP8. This feedback will especially be taken into account for incremental improvement and extension of models and technologies. The technical integration takes place jointly with WP7.
The technical specifications, workflows and information model developed by WP7 provide the context for integration of WP4 components into the overall system.

The deliverable itself will be complemented and extended by D4.2 (Second Release of Text-analysis Technologies and Models) in PM28 as well as by D4.3 (Final Release of Text-analysis Technologies and Models) in PM 36.

2 Language Identification

Language identification (LID) addresses the problem of assigning a natural language (or languages) to a document of part of a document. In a broad sense, this applies to input of any modality, such as audio, image (sign-language) or text. Within the scope of MIRROR, only input in textual form is targeted. In many instances, the language of a text might be known, e.g. by information available about the source or meta-data provided by the supplier. In other cases, the language may not be known in advance and needs to be determined before being able to proceed with further processing steps.

2.1 State of the Art

The identification can take place within a (pre-defined) closed set of languages or an open set of languages and may also target dialects. Furthermore, LID may be applied to texts exhibiting standard forms of spellings as well as to highly informal texts from Social Media, resembling more spoken language rather than written language in many aspects. LID has been extensively researched over the last decades. Jauhiainen et al. (2018) provide an excellent overview of LID from its beginnings to the recent developments and state-of-the-art approaches. LID forms an integral part of many NLP processing chains, as downstream models for enrichment often are language-dependent and require an accurate classification of input before being able to process it.

Work on LID has traditionally focused on mono-lingual documents and produces classifications on the document level. However, multilingual documents may be processed by appropriate segmentation (into language-homogeneous sections) and application of LID to these sub-sections. Social Media, frequently exhibiting only short texts of an informal style and a including mix of user/account names, hashtags and URLs pose especially challenging settings to LID.

2.1.1 Evaluation and Metrics

As for many other NLP tasks, the evaluation of LID systems is typically carried out in terms of precision, recall and the resulting F-measure. This requires a set of reference documents, whose language has been determined by human annotators. Evaluation typically takes place on the level of the full document. Performance typically depends on the length of the text to be classified and increases with length. As typically a whole set of languages form part of the inventory of languages to be identified, results are frequently presented in the form of a confusion matrix, allowing to determine error-patterns across different language pairs (Jauhiainen et al., 2018). Results may be calculated by micro-averaging or macro-averaging elements of the test-set, assigning different importance to the number of documents per language if these are not identical. Whereas earlier work focused predominantly on English and selected Western languages, the set
of target languages has constantly been extended and lead to evaluations of more than 1300 languages (Brown, 2014). Such settings owe their existence to the ever increasing availability of (digital) texts in more languages and dialects as well as to an increase in interest and (commercial) applications requiring robust LID. However, also in this setup low resource languages exist, such as Nepali, Urdu or Ukrainian (Bergsma et al., 2012).

### 2.1.2 LID Domains and Challenges

The domains of LID are manifold and have produced LID systems of various scopes and target languages. Automatic translation, Web-browsing, document retrieval, query engines, text-based research and filtering form just a few of the application domains which have been targeted by LID systems over time. Many NLP components require initial application of LID to determine the language of their input, as they typically work as mono-lingual systems only. Cleaning, tokenization, NER (see chapter 4), automatic summarization or topic detection present some of these technologies (see (Jauhiainen et al., 2018) for a more extensive list of applications of LID).

Whereas the performance of LID systems has increased dramatically over the past decades, several challenging areas for LID still remain. In a (large) set of languages, the performance per language often differs substantially across the individual languages. This may be due to an unequal amount of training data being available, issues related to morphology or domain specificity of texts. The brevity of texts (especially on Social Media), appearance of special characters, hashtags, unknown language, mixes of languages and potentially even interspersed text of URLs and computer language code (e.g. in the context of web-crawling) have been identified as further problems for LID. The particular challenges and the need for distinction between similar languages has been addressed by the Discriminating between Similar Languages (DSL) shared task by the VarDial workshops held at COLING between 2014 and 2017 (Zampieri et al., 2014; Zampieri et al., 2015; Malmasi et al., 2016; Zampieri et al., 2017).

### 2.1.3 LID Approaches

Since LID has been an active field of research for several decades, a wide variety of methods has been investigated within its scope. This applies to the set of features used as well as to the kinds of classifiers trained and evaluated using them.

Regarding the features, bytes, byte-encodings, characters, character combinations (character n-grams), morphemes syllables, chunks, words, word-combinations have been used in isolation or in combination together with a series of feature smoothing mechanisms (Jauhiainen et al., 2018).

Many types of classifiers, based on individual features or combinations of features have been investigated, such as decision trees, discriminant functions, Support Vector Machines (SVMs), Naive Bayes, or graph-based n-grams (Mathur et al., 2017; Jauhiainen et al., 2018). In more recent setups, ensembles of classifiers have been investigated, yielding state-of-the-art results.
2.1.4 Performance and SOA

Performance of LID systems has been approaching that of humans over the past years and to some extent may already surpass it. Especially in large sets of languages, human knowledge about these languages may be limited and a single person may not be able to distinguish (as accurately as an algorithm) between these languages. The performance and state-of-the-art depends heavily on the number and kinds of languages, the type and the length of text used for evaluation. Accuracy on large sets of languages (285) and 70 character length test texts have been reported to be 60% (Jauhiainen et al., 2017) whereas results on the identification of 6 languages over tweets of an average of 80 characters has been reported to be 99.8% (Vogel and Tresner-Kirsch, 2012).

Even though the field has advanced both in terms of features as well as in terms of classifiers, the methods originally investigated by Cavnar and Trenkle (1994) can still be regarded as (almost) state-of-the-art – and certainly as a strong baseline for reference (on an evaluation set covering newsgroup-content in 9 languages they report between 92.9% and 99.8% accuracy). Their seminal work in combination with an implementation called TextCat\(^1\) have made this method an extremely popular one within the arena of LID. It is based upon the creation of document- and language-specific sets of character n-grams. The most frequent n-grams are ranked for both the language as well as a document to be classified and the difference in ranks is accumulated, producing the language with minimum rank-based-distance as the classification of the document.

Evaluations are heavily dependent on the number and variety of languages used as well as the quantity of training data (per language) available. A number of studies have therefore been devoted to the issue of what effect the amount of training data has on accuracy and how long texts have to be before their language can be identified correctly. Especially in the context of classifying large amounts of texts, performance in terms of throughput and complexity become increasingly important and need to be taken into account for evaluation and comparison of results.

2.2 Relevance to and Approach within MIRROR

As stated above, robust LID is an important foundation for a variety of downstream enrichment tasks, such as NER, summarization or translation. LID also holds this place within the processing framework of MIRROR. Within the Media Mining System – which serves as a main source of content for MIRROR – the language of a document is determined in different manners. A version of the method devised by Cavnar and Trenkle (1994) has been implemented within the Media Mining System. The language and origin of TV, radio and Web-based content and sources is assumed to be known (and has been verified). For sources where this is not the case, but for which the language is provided as part of the meta-data of a document (e.g. for Twitter), the above component is used for verification against the platform. Cases where the classification differs are marked; the LID of the component is used for further processing. For platforms, which do not provide LID-information, the component is used to determine the language of the document. Since it cannot be guaranteed that the language of a document is among a set of previously fixed set of languages, an open-set stance must be assumed (that is the result of classification is that the language is among the given set or some other,

\(^1\) https://www.let.rug.nl/vannoord/TextCat
unknown, language). To be able to determine the language of textual content ingested into the MIRROR framework which does not enter through the MMS, the LID component will be packaged into a container and become part of the workflow for text documents.

### 2.3 Component Integration

The LID component will receive text as input (payload of a REST request) and provide a ranked list of languages (code and score) as output.

**Input:** “Migration has been a hot topic across Europe in the past.”

**Output:**

```json
{
    "response": {
        "type": "classification",
        "classes": [
            {
                "class": "en",
                "score": 0.78
            },
            {
                "class": "cs",
                "score": 0.45
            }
        ]
    }
}
```

### 2.4 Models/Technologies/Insights available/under Development for MIRROR

The current implementation of LID will be integrated into the MIRROR framework and workflows. It is planned to investigate to what extent multi-language documents (resp. documents containing words and expressions in multiple languages) have on LID. The segmentation of multilingual documents into language-coherent sections with subsequent classification and section-dependent enrichment form further alleys of future work.

### 3 Extraction / Cleaning / Tokenization

Within the context of MIRROR, textual content may be extracted from sources and platforms using an API or be ingested from files. Processing entails a series of steps to convert input into a form which is processable by downstream technologies. This includes the removal of extraneous characters (such as markup, formatting information or even pieces of programming code), special handling of punctuation or numbers, abbreviations or quotes, cleaning and breaking up the input text into a sequence of words (or tokens) forming the units of further processing. Depending on the type of language and writing system, words may be delimited by spaces or be part of longer strings requiring segmentation (e.g. for agglutinative languages or languages not using space to delimit words like Chinese or Thai). Consequently, cleaning and tokenization typically combine language-independent and language-dependent processing steps.

#### 3.1 State of the Art

The extraction of textual data and subsequent processing (e.g. eliminating from this text editorial and navigational information, commercials or text which is not part of the article itself) is a common task for the
processing of contents from the Internet. In many instances, producers of news offer access to structured content via RSS\(^2\) or ATOM\(^3\) feeds or more specialized markup-languages such as NewsML\(^4\). In other cases, Web-Pages in HTML are the only available source. The latter case requires specific processing for the extraction of relevant text and elimination of irrelevant sections (such as navigational content or advertisements etc). Kohlschütter et al. (2010) describe the challenges of this setup in more detail and suggest several approaches for the elimination of so-called boiler-plate text. (we include such processing and further processing based on regular-expressions under the term cleaning). Social Media platforms typically provide content in a (semi-) structured form, allowing to extract relevant text from within structures returned by the respective API. More recently, news-producers have been using Social Media as an additional distribution channel, e.g. by tweeting about articles and linking back to them via an embedded URL. These tweets typically have the structure of a headline with an associated URL. Over the past few years, Social Media platforms have been changing their APIs and terms of use multiple times. Rate-restrictions and an increase of the effort to register are common-place. More recently, the provision of a video, displaying the use of data retrieved via an API has become mandatory for FB and YouTube. Concerning RSS- and ATOM-feeds, the maintenance (and sometimes even) the correctness of feeds requires ongoing attention: providers change endpoints of feeds or take feeds offline for a period of time or permanently. Changes in terms of use as well as API changes are often only announced after the fact or can only be detected by scanning of developer-fora.

The term tokenization is used to describe the processing and partitioning of text into meaningful units (or tokens). Tokenization is a fundamental step typically applied during early stages of processing. Approaches include simple tokenization into individual tokens by separating words delimited by spaces or more elaborate approaches for languages and writing systems which do not employ spaces to delimit words. Such cases include agglutinative languages (e.g. Turkish), languages exhibiting composition (e.g. German or Dutch) or languages which use sequences of characters with limited spacing or without spacing (e.g. Mandarin or Thai). Depending on the desired granularity and task, multiword tokens, where a single orthographic token corresponds to multiple words (such as in clitics, e.g. Spanish: dámelo = da me lo or contractions, e.g. French à le-> au, abbreviations and acronyms) form further areas of tokenization processing. For many tasks, simple tokenization taking into account language-dependent morphology is considered sufficient.

### 3.2 Relevance to and approach within MIRROR

The MIRROR platform is envisioned to process texts from a variety of traditional and Social Media sources. Wherever possible, publicly available APIs and standard interfaces will be used for this purpose.

Content from traditional sources such as newsfeeds from newspapers, magazines, journals, think-tanks, scientific and policy-making institutions will be collected and processed on a regular schedule or on demand. The SAIL LABS Media Mining System (MMS) currently offers more than 7000 such sources in 35 languages

---

\(^2\) https://www.rssboard.org/rss-specification

\(^3\) https://tools.ietf.org/html/rfc4287

\(^4\) https://iptc.org/standards/newsml-g2
Deliverable D1.3

from around the globe. This set of feeds and content provides the basis for content production; it can be accessed by the MIRROR platform through an API.

The MMS likewise comprises so-called collectors for a variety of Social Media platforms: Twitter, Facebook, YouTube and VKontakte. These collectors are adaptors to the respective APIs offered by platform providers and comprise search (request/response) and streaming APIs.

Both, the infrastructure for the collection of traditional as well as of Social Media sources has been in operation for several years and was made available as base-infrastructure to MIRROR by SAIL LABS in order to be able to access contents from day one of the project.

However, neither the traditional nor the Social Media sources and their access and processing can be regarded as static. Both types of sources require constant monitoring as well as adaptation when APIs change. During the first year of MIRROR this has led to a number of changes, e.g. to the interfaces of Facebook and YouTube regarding authentication, the availability of certain fields or access-quotas.

The assignment of statements to certain regions (countries) is a fundamental requirement when wanting to compare the coverage of topics between different countries (e.g. assigning a statement concerning the healthcare system of Austria to Turkey as the provenance of this statement). Within MIRROR, this is essential for comparison of topics and their coverage in countries-of-origin (COO), countries-of-transition (COT) and countries-of-destination (COD). In the first year of MIRROR, properties regarding origin has been attached to all sources collected from potential regions of interest to MIRROR: Central Asia (Pakistan, Afghanistan, Iran), Western Asia (Iran, Iraq, Syria, Jordan, Lebanon, Turkey), Europe (EU, Russia, Balkans). Furthermore, the previously diverse naming scheme with the MMS has been corrected to follow standard naming schemes5.

The initial stages of processing of textual content (cleaning, segmentation, tokenization6) have been extended with heuristics collected during inspection of content. These heuristics are applied for the segmentation of textual content and the filtering of relevant sections (e.g. paragraphs need to exhibit a certain number of connected sentences each exhibiting a certain minimum number of tokens to be accepted, the ratios of non-character, number, and special-characters are taken into consideration). This has led to a refinement of the previous process now termed smart-cleaning. For unsupervised processing and creation of ASR models, this is deemed advantageous as this scheme improves the text and words considered for model update.

**Progress made within MIRROR**

- Assignment of origin to sources for localization of content
- Monitoring and maintenance for sources in geographic regions relevant to MIRROR
- Extension of cleaning and tokenization algorithms based on heuristics (smart cleaning)

---

5 Both, the localization as well as the naming-scheme will be consolidated with work performed in WP6.

6 Cleaning and tokenization will be performed within the respective components rather than by separate components. This will be achieved by re-use of PERL modules for these components.
3.3 Models/Technologies/Results/Insights/Next Steps

The assignment of origins to all sources for countries relevant to MIRROR\(^7\) has been completed and now provides a sound basis for localization of topics, statements etc concerning migration relevant topics. The extension of the initial text-processing stages has been deployed. The results of processing are collected and transmitted via email to employees within SAIL LABS for manual inspection (as a sanity check and to be able to detect errors for further refinement).

These ongoing tasks will also form future work of MIRROR. We expect further (and sudden!) changes especially with regard to Social Media API which will require adaptations for collection. The addition of further sources depending on end-user feedback and information obtained from field-tests is likewise expected for the future. On a technical level, containerization of the components will form an area of activity. This will allow the operation of these components within the MIRROR framework and decrease the dependency on the MMS (as the MIRROR system is eventually envisioned to be completely independent of the MMS).

4 Named Entity Detection / Concept Detection

With the phenomenon of digitization and the invention of the Internet, an abundance of textual information has become available on a variety of sources, spanning numerous domains and diverse styles, genres and languages. These texts come in unstructured form, different formats and are produced on a massive scale by professionals as well as individuals. They typically contain entities with similar properties, referring to semantic types and carrying crucial information about the content. These entities can be *rigid designators*\(^8\) or more *abstract entities*, not referring to any particular designator and rather representing *concepts*\(^9\). Within the scope of this work, no particular distinction is made between these two types unless when explicitly stated and both kinds are referred to as entities. Entities provide valuable information about the content and provide key elements for further analysis. Their detection is thus an important step in the enrichment chain and serves as the basis for further technologies.

4.1 State of the Art

The term *Named Entity* (NE) was coined by the 6\(^{th}\) Message Understanding Conference (MUC-6 ) (Grishman and Sundheim, 1996) in 1996 to refer to word-forms having similar properties. Whereas MUC focused on

---

\(^7\) As far as this has been determined. All further remaining sources are currently also being assigned origins.

\(^8\) A term is said to be a *rigid designator* or *absolute substantial term* when it designates (picks out, denotes, refers to) the same thing in *all possible worlds* in which that thing exists, https://en.wikipedia.org/wiki/Rigid_designator

\(^9\) We regard UNHCR, the UN Refugee Agency as a *Named Entity* whereas asylum seeker or refugee influx would be regarded as *concepts*. Both kinds are processed in the same manner via the technologies described in this chapter.
Information Extraction (IE) (Pascua et al., 2006), similar downstream tasks such as question-answering\(^{10}\), machine translation (MT), summarization of the population of knowledge graphs can likewise benefit from NER. During the initial phases of NER, particularly for MUC, it was deemed essential to recognize information units such as the names of persons, organizations, locations as well as a set of numeric expressions such as dates, time, amounts of money and percentages prior to performing information extraction. These units (sometimes also called tags) formed the standard set of Named Entities for years to come and are still of relevance today. The task of identifying them has become known as \textit{Named Entity Recognition and Classification} or NERC. As this name suggests, the task is two-fold:

- detecting/recognizing Named Entities (Named Entity Recognition, NER) and
- classification of such a detected NE into one or several predefined-classes (commonly only one class)

The task of NER can formally be defined as follows. Given a sequence of (input) tokens \( s = <w_1, w_2, ..., w_n> \), NER is to produce a set of of tuples \(<l_b, l_e, t>\) each of which corresponds to a particular NE mentioned in \( s \). \( l_b \) and \( l_e \) are the begin and end indices respectively in \([1..n]\) while \( t \) denotes the type of NE (from a pre-defined set). NERs may stretch over several tokens or be limited to a single token (a singleton). The start and end index may thus be identical for single-token cases.

Depending on the exact setup and domain, the granularity of entities to be detected can be distinguished into coarse (person, organization, location, number) versus fine-grained entities (stadium, drug, disease, airplane, weapon) (Ling and Weld, 2012).

The following sample about the current (2020) president of the United States contains several NEs:

\[
\text{Donald John Trump was born in Queens, New York.}
\]

\[
\begin{array}{|c|}
\hline
\text{<w$_1$, w$_3$, Person>} & \text{Donald John Trump} \\
\hline
\text{<w$_7$, w$_7$, Location>} & \text{Queens} \\
\hline
\text{<w$_9$, w$_{10}$, Location>} & \text{New York} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
\text{Named Entity Recognition} & \text{Donald} & \text{John} & \text{Trump} & \text{was} & \text{born} & \text{in} & \text{Queens} & \text{New} & \text{York} \\
\hline
\text{w$_1$} & \text{w$_2$} & \text{w$_3$} & \text{w$_4$} & \text{w$_5$} & \text{w$_6$} & \text{w$_7$} & \text{w$_8$} & \text{w$_9$} & \text{w$_{10}$} & \text{w$_{11}$} \\
\hline
\end{array}
\]

\textit{Table 1 NER example}

In the tokenized text (see chapter 3), two locations and one person can be recognized (tagged) in this sentence. Additional Entity Disambiguation (also referred to as \textit{Entity Linking}) may be required. In the above case this disambiguation will determine that New York in the sentence above refers to the city rather than to the state.

\(^{10}\text{According to Guo et al., (2009), 71\% of all search queries contain at least one NE}\)
Following MUC, a series of events and evaluations was staged: Information Retrieval and Extraction (IREX), 2000 (Satoshi and Hitoshi, 2000), Conference on Natural Language Learning (2002 and 2003) (Sang, 2002; Sang and Meulder, 2003), Automatic Content Extraction (ACE) (Doddington et al., 2004) and HAREM (Santos et al., 2006) all bearing witness to the continuous and growing interest in this field.

Several of these programs and evaluations led to a series of datasets (some of which are available as open-source, see Li et al., 2020 for a comprehensive overview) as well as to a standard evaluation methodology; both of which are essential in allowing others to reproduce and to compare results.

4.1.1 Evaluation and Metrics

The performance of NER is typically evaluated against a reference created by human annotators (who need to possess the appropriate language skills as well as the expertise regarding the particular domain).

Following the definition given above, NER is comprised of two basic sub-problems: the detection of the boundaries of a NE and the identification/classification of its type. Evaluation strategies differ between exact and relaxed matching of these elements.

In Exact-Matching evaluation (Tjong et al., 2003), the boundaries and type of a NE are evaluated simultaneously and a correct match is only credited if both measures are correct.

In Relaxed-Matching evaluation (Grishman and Sundheim, 1996), a correct type is credited if an entity is assigned its correct type, regardless of the boundaries, as long as there is an overlap with the reference. On the other hand, a correct boundary is credited regardless of the entity’s type assignment. More complex evaluation methods have been introduced (e.g. by the ACE evaluation, Doddington et al. 2004) but have failed to gain wide-spread acceptance and adoption.

In both cases, measures like precision, recall and (their harmonic mean) the F-measure are used. Whereas in the exact matching case above, the true and false positives and negatives are defined on the individual NEs themselves, in the latter case the score takes into account both the type as well as the extent of the NEs individually (Nadeau and Sekine, 2007). Micro-averaging (calculating the F-measure for each type of NE individually and then computing their average) as well as macro-averaging (computing the F-measure over all types together rather than individually and then averaging the resulting values) has been used to determine the performance of NER. Micro-averaging was employed by MUC.

4.1.2 NER Domains and Challenges

Several factors have an impact on the creation and performance of NER systems and models.

Language: whereas early efforts concentrated on English, other languages have also received attention over time. German, Dutch and Spanish have been targeted during the CONLL conferences; Arabic received ample attention in ACE and Portuguese was studied extensively in HAREM. Mandarin Chinese has received wide attention during recent years. Further languages have received some (reduced) attention such as Turkish, Burmese, Vietnamese, Mongolian, Polish, Russian, Czech, Malay, Bahasa Indonesia, Mongolian, Urdu and...
Indian languages such as Hindi, Gujarati, Bengali and Manipuri (for references see Goyal et al., 2018 and Li et al. 2020).

**Genre and Domain**: whereas initial work on NER focused predominantly on the news-domain (WSJ, Reuters, NYT, business news, etc.), specialized (bio-) medical domains (genes, gene-products, health-records, TCM, etc.) have received considerable attention during the last decades. With the rise of Social Media in the first decade of the 2000’s, an increased interest in tagging NE in unstructured texts and short messages can be noted. The typical use of colloquialisms, slang, typos, different language resisters and code pose additional problems for NER typically not encountered in traditional media.

Entity Types: triggered by MUC, a limited and coarse set of entity types (tags) was used in early work. Several evaluation campaigns introduced further unspecific (class: miscellaneous) or specific entity types (weapons, vehicles, proteins, chemicals,…). The latter are also referred to as fine-grained entities.

**Nested Entities**: entities embedded within other entities (such as the New York Giants) provide a challenge to NER systems. Different kinds of segment representation techniques have been devised to counter this effect. Likewise, the possibility to assign multiple classes to (overlapping) entities might be a viable solution.

Ambiguity: strings may appear as a NE in certain locations and as proper nouns in others or they may be of a different type depending on their context (*Paris Hilton* (the person) vs *(at the)* *Paris Hilton* (location)). NE disambiguation provides an approach to determine the correct type based on the context.

**NE-Corpora**: annotated training (and evaluation) data is a requirement for the development of NER systems based on supervised methods (evaluation for all methods). Producing annotated references is tedious and costly and requires human labor, often needing expert skills (language, domain) for correct annotation. Measuring inter-annotator agreement in addition increases the effort required to establish such corpora. Even if annotated corpora do exist, the question remains as to whether the set of NE (tags) and their granularity corresponds to the task at hand.

**Technical Resources**: in addition to the scarcity of data (corpora), a lack of technological components might exist. Many components (POS-taggers, stemmer, morphological analysis,...) may only support a limited set of languages. The individual support of languages may differ dramatically (just having a certain language listed as “supported” has often been found to be a mere exaggeration).

### 4.1.3 NER Approaches

The methods applied to NER can be broadly divided into four approaches:

- Rule- and pattern-based
- Unsupervised Learning
- Supervised Learning
- Deep Learning

The above approaches also correspond to the development of NER systems over the past decades, with rule- and pattern-based systems laying the foundations and DNN based methods being the state-of-the-art. However, not all NER systems are created equal and as such different systems may exhibit strengths and weaknesses in different domains and application areas.
Rule/Pattern-based

These systems depend on a set of manually crafted rules and patterns, domain specific lists (such as gazeteers) and syntactic/semantic patterns to recognize entities. Due to the rigidity and domain-specificity of rules, incompleteness of lists, high precision and low recall are often observed from such systems (Li et al., 2020). In addition, these systems are domain specific and cannot be ported easily to further domains.

Unsupervised Learning

Mentions of NE are inferred by the application of clustering over large corpora of texts. NEs are then extracted from clustered groups depending on their common context. An initial set of seed rules/patterns may be used for guidance in the process; Collins and Singer (1999) demonstrate this approach using a mere 7 rules as seeds for their work.

Supervised Learning

Supervised methods use a corpus of annotated NEs and employ various types of machine learning (ML) algorithms to learn a model which is able to recognize NEs in similar but yet unseen data. Feature-engineering (Zhang and Casari, 2018) is a central element in these approaches; a wide array of word-level (case, morphology, POS), list-lookup (gazeteers, DBPedia\(^{11}\)) and document-level or corpus-level features (multiple occurrences) has been employed for this task. Based on these features, a variety of ML algorithms have been applied, such as Hidden-Markov Models (HMMs), decision trees, Support Vector Machines (SVM) or conditional random fields (CRF). For a comprehensive overview see Goyal et al., 2018). Supervised methods rely on the existence of corpora – which may be a bottleneck for certain languages and domains.

Deep Learning

Deep Learning (DL) and Deep Learning based NER (DLNER) has gained wide attention over the last years and is becoming the dominant method for achieving state-of-the-art results for NER. DL is a sub-field of ML which employs multiple layers of processing to learn representation on multiple levels of abstraction (LeCun et al., 2015). Its key advantage is the capability of representation learning and semantic composition empowered by both, vector representations and neural processing. DLNER benefits from non-linear transformations, generating mappings from input to output and effectively eliminating the need to specific features (and feature engineering). Li et al. (2020) categorize DLNER systems according to three dimensions: the distributed representation of the input (word and character embeddings, POS-tags, being elements of lists such as gazeteers,…), the context-encoder, capturing context dependencies using CNN, RNN or Transformers networks and the tag-decoding stage using CRF or RNN, predicting (properties of) tokens in the input sequence.

Goyal et al. (2018) provide a comprehensive overview of approaches used (Table 2 for rule/pattern-based systems, Table 3 for ML based methods). Li et al. (2020) provide a list of “off-the-shelf-tools” for NER offered by academia and industry.

\(^{11}\) https://wiki.dbpedia.org
4.1.4 Performance and State-of-the-Art

As noted above, not all NER systems are created equal and this is also true for the type and scope of tasks which NER have been applied to. Dramatically different setups and datasets frequently make comparisons of individual systems impossible. Whereas some systems are evaluated on standard test-sets in English, others are evaluated against more exotic settings such as biomedical data in Hindi\(^{12}\). The language, domain, availability of resources and features, the number and kind of entities and the way these are scored provide a wide range of NER applications and results in widely differing outcomes.

Rule-/pattern-based approaches are regarded to provide relatively good performance due to the fact that they follow language-specific rules and language-specific resources such as gazeteers, POS-taggers and language-specific morphological processing. This performance is offset by a high system development cost and comes at the expense of not being portable to other domains or other languages. Such systems typically exhibit high precision and low recall (Li et al., 2020). F-measures reported in the literature range from just over 22% - for user-generated content concerning consumer electronics and automotive domains – to over 91% for NER of the standard set of NE in Urdu. F-measures in the high 80’ies are quite common across the board. For an overview see Goyal et al. (2018), Table 2.

Supervised learning approaches provide slightly better performance than rule/pattern-based approaches but are highly dependent on the availability of labelled training data. A variety of techniques and models have been used within this line of research; CRF, Naïve Bayes, LDA, k-NN are some of the methods applied within the supervised paradigm, yielding F-measures from 52% to over 93%.

Unsupervised and hybrid approaches are based on clustering of large resources (e.g. Wikipedia) to arrive at similar contexts of entities. Hybrid approaches combine several classifiers and yield F-measures of up to almost 95%.

More recently, DL-based approaches have successfully entered the NER-stage and met with success. Several recently created systems (since 2018) employ a combination of hybrid character- and word-embeddings as the input representation, LSTM (Long Short-Term Memory), RNN (Recurrent NN) or GRU (Gated Recurrent Unit) in the context encoding stage and CRF or SoftMax-Layer for the tag-decoding (ie. assigning tags to tokens in the input). Embedding-approaches benefit from models made available by research organizations or large companies which have been trained on enormous amounts of data, on huge computing networks and still requiring weeks of processing time. As these resources are typically only available to a select few and tasks often target domains different from those which these models were trained on, the idea of transfer learning is gaining momentum – adapting large pre-trained models with smaller amounts of domain-dependent models. F-scores obtained by SOA systems on OntoNotes5.0\(^{13}\) task reach 92.07%, whereas on CoNLL03 the best F-measure obtained is 93.5%. Regarding data from informal sources (social media) the best F-measures currently reached are slightly above 40%.

\(^{12}\) This is by no means meant to downplay efforts, results or sound pejorative in any way!

\(^{13}\) https://catalog.ldc.upenn.edu/LDC2013T19
4.2 Relevance to and Approach within MIRROR

As outlined above, NER plays a central role as a pre-processing step for further downstream IE processing, such as summarization, sentiment analysis or network analysis. The input for NER within MIRROR is composed of textual data from a wide spectrum of sources\textsuperscript{14}, a variety of channels and multiple languages and language-styles (in the sense of registers, colloquialism, special terminology, etc.). All of these factors pose challenges to a NER system and influence its performance. The wide range of results obtained by state-of-the-art systems described above demonstrates that there is no one-system-fits-all approach possible which would yield high performance across all the different kinds of input and thus a rather large variance in performance should be expected. The output of NER within MIRROR will be used for analysis and visualization purposes per se (e.g. detection of cities in countries involved in migration processes) as well as in the processing of the Migration Related Semantic Concepts (MRSCs). These form a central element in the detection of perceptions (and mis-perceptions) and hence in the assessment of possible threats; the latter being the ultimate goal of the MIRROR project. MRSCs are expected to correspond to latent topics with associated sentiments whose general factors have been described in D2.1. Economic, social, demographic, environmental or legal factors will form core issues of the MRSCs. To this end, the NE within MIRROR will include the standard-set of entities (persons, organizations, locations) but also go beyond these for issues related with the above factors. Geographic entities will be linked on a provincial as well as a country level in order to allow for better aggregation regarding country-of-origin (COO), country-of-transition (COT) and country-of-destination (COD) analyses. Entities from areas corresponding to the above factors such as ethics and religion, healthcare, education, role of genders will be developed with the aim to allow the detection of and to provide support for these high-level issues. Furthermore, all work will be carried out in the languages relevant to the MIRROR project whenever possible. This will comprise languages spoken in COO’s, COT’s as well as COD’s as identified in the use-cases and the requirements of end-users within the project.

Progress made within MIRROR

- Extension of entity-classes to migration relevant ones, addition of concepts related to MRSCs for multiple languages for regions relevant to MIRROR, e.g. Remittances, Xenophobia, Human Trafficking, Refugee Camp, ...)
- Creation of multi-lingual (for languages relevant to MIRROR) entries these concepts

4.3 Component Integration

The NER component will receive text as input (payload of a REST request) and provide a list of annotations of detected entities (type, start and end positions, tokens) as output.

Input: “Migration has been a hot topic in Germany and Sweden this year.”

Output:

```json
{
    "response": {
        "type": "texts"
    }
}
```

\textsuperscript{14} These texts also undergone prior processing such as cleaning and language identification
"texts": [
{
  "content": "Immigration has been a hot topic in Germany and Sweden last year.",
  "annotations": {
    "migration": [{"start": "0", "end": "9", "features": {"text": "immigration"}}],
    "location": [{"start": "25", "end": "32", "features": {"text": "Germany"}}],
    "location": [{"start": "34", "end": "41", "features": {"text": "Germany"}}],
    "location": [{"start": "25", "end": "32", "features": {"text": "Sweden"}}],
    "location": [{"start": "37", "end": "43", "features": {"text": "Sweden"}}]
  }
}
]

4.4 NER Models and Technologies available/under Development for MIRROR

The current implementation of the NER component (by SAIL LABS) is based on a rule-/pattern-based approach. However, the component also allows for a feature-based mode of operation, based upon an HMM-based NER approach originally proposed by (Bikel et al., 1997). The rule-/pattern-based approach was selected due to its robustness and the fact that it allows matching and linking of NEs across languages. The absence of training data for many of the targeted languages lends further support to this approach. The inventory of the rule-/pattern-based system was created by processing of DBPedia and Geonames15 and augmented over time by additional proprietary datasets and manual addition by domain experts.

Within the SAIL LABS Media Mining System (MMS), a broader approach to NE has been chosen insofar as the inventory and classes extend well beyond the classical ones (and covering classes which were not found among those used by any of the state-of-the-art papers). A set of almost 240k entities divided into more than 20 categories currently provides the NER support for content in over 30 languages. All NEs are identified by a unique ID (URI) which allows for cross-lingual linking of NEs. Furthermore, different surface forms can be identified with the same NE. The NER component provides pre-processing and some basic morphological-processing (to be able to handle inflection and affixes). In addition, basic co-referencing of NEs within documents is supported.

As the MMS serves as one of the data-sources for the MIRROR project, tagged NEs are part of the information exported from the MMS and ingested into the MIRROR framework. In addition, the NER component will be made available as a separate component within the MIRROR workflow(s). As such, it will be possible to perform NER on further textual data ingested into the system.

The classes of NE available for MIRROR16 can be divided into two groups:

- Standard set of persons, organizations and locations
- Domain specific sets of economic, social, environmental or legal issues

---

15 https://www.geonames.org

16 On the one hand, a large set was available in the MMS before the MIRROR project; on the other hand, the set of NE is constantly being expanded by domain experts and current events in the (migration related) fields relevant to MIRROR.
The former set is expected to play a role when attributing and localizing information (what/who/which location or organization or individual is talked about or mentioned). The latter set is expected to be of importance regarding migration-related topics and expectations expressed in content.

### 4.5 Next Steps

The current set of NEs will be continuously extended by monitoring of news and events concerning migration. Based upon analysis of a selected set of documents, an analysis with regard to collocation of words will be carried out. This may provide insights on multi-token expressions which could be considered as separate entities. Furthermore, experiments on word-embeddings are planned with the goal of detecting semantically related words which could form the basis of further NEs. However, when pursuing these lines of research, the focus will remain on techniques allowing the multilingual tagging of content and/or the rapid transfer of methods to further languages (without requiring resources which may not be available in the languages relevant to MIRROR).

No formal evaluation of the NER component has taken place at this point. This is primarily due to the lack of an adequate test-set representing the domain and languages of the MIRROR project. It is planned to examine the publicly available test-sets for fitness of use for our domain and perform evaluations on these test-sets or to create such a test-set (spanning multiple languages) within the MIRROR project itself as part of the work in year two.

### 5 Sentiment Analysis

Sentiment Analysis addresses the problem of determining the emotional state(s) expressed by an author in a given text and aims to determine its polarity. Within the process, polarity may be classified into positive and negative, sometimes with a third class added for neutral or even a fourth class for mixed. The classes may further be subdivided into categories for strong or weak expression of the respective sentiment or even into a range of categories like cold anger, sadness or joy or other fine-grained emotions. The latter terms are sometimes also referred to as emotion tones.

#### 5.1 State of the Art

Esuli et al. (2006) categorized SA approaches into 3 classes identifying text SO-Polarity (Subjective/Objective), text PN-polarity (Positive/Negative) and into strength of PN-polarity. Further approaches refer to these elements as arousal and valence (Russell, 1980) or as personality traits (Kassin, 2003). No single coherent terminology exists which can be ascribed to the fact that SA has its roots in various fields such as psychology, economic sciences, computer science, natural language processing (NLP) or linguistics all with their proprietary naming schemes and heritage (and potentially disjoint research communities). Within the fields of computer science and NLP, the classification into positive, neutral and negative, frequently with an

---

17 We only explicitly address text as input here but acknowledge that it is likewise possible to detect emotions and sentiment from audio, visual or combined content.
associated symbolic (very, weakly) or numeric marker for intensity (positive on a scale of +1 to +5) to arrive at a combined classification, are commonplace. We will limit our discussion to the points of view from NLP and computer science and accordingly regard SA from an angle of a multiclass classification problem.

5.1.1 Evaluation and Metrics

Performance of SA systems is commonly measured in terms of accuracy against a reference. An evaluation set is tagged with the respective portfolio of annotations and accuracy is subsequently measured against the reference-tagging of the same set in terms of precision and recall and the resulting the F-measure. References may be annotated on a number of levels, such as clause, sentence, paragraph, syntactic-structure or document. In addition, only certain aspects of statements may be tagged for reference. In particular in case of Social Media (SM), content may be short and the SM-post as a whole may be the unit for annotation and evaluation. The creation of such references in many circumstances is not a trivial task, as classification into a set of categories by human annotators depends on their cultural background, political views, personal experiences, emotional state, etc. Because of this inherent problem, annotation by multiple experts is a common approach in the field (Shalunts and Backfried, 2015). Even in the case of a two-class problem, inter-annotator-agreement may not be 100%. However, the measured inter-annotator-agreement18 may be employed as a desirable level of performance when evaluating an automatic classifier.

5.1.2 SA Domains and Challenges

There has been a substantial amount of activity and interest in SA over the past decades. The most common applications are the monitoring of public opinions in marketing (product reviews), entertainment (movies), politics (election campaigns) or the detection of potentially dangerous activity on the Internet (Forsyth and Martell, 2007). The application fields of SA are innumerable and so it has been applied in diverse fields, such as customer feedback, disaster management, airline services, mobility network evaluation, healthcare, customer relationship management (CRM), or banking and insurance1920. Within commercial applications, SA is also frequently referred to as Emotion Detection. It has been introduced in tools and applications such as IBM Watson Tone Analyzer21 or Microsoft’s Azure Cognitive Services22 (and many others).

As in many other sectors of NLP, work on SA has been focusing on content in English. However, SA has also been applied to contents in several other languages such as Spanish (Perez Rosas and Banea, 2012; Moreno-Ortiz and Hernandez, 2013), French (Balahur and Turki, 2012), German (Remus et al., 2010), Russian (Chetviorkin and Loukachevitch, 2013), Turkish (Demirtas and Pechenizkiy, 2013) or Bahasa Indonesia

18 Typical measures include Cohen’s Kappa or Krippendorf’s Alpha, for an overview see (Artstein and Poesio, 2008)
19 https://www.sciencedirect.com/topics/engineering/sentiment-analysis
20 https://www.sciencedirect.com/topics/computer-science/emotion-detection
22 https://docs.microsoft.com/en-us/azure/cognitive-services/
(Shalunts et al., 2017). Furthermore, work has been carried out on the combination of machine translation (MT) and SA which may provide a viable alternative for circumstance when no language-specific SA system is available (Shalunts and Backfried, 2016a).

Several test-sets have been produced within the scope of evaluation campaigns such as OpeNER (hotel reviews) (Agerri et al., 2013), SemEval 2013 (tweets) (Nakov et al., 2013), MPQA (news) (Wilson et al., 2005), the Stanford Sentiment Treebank (movie reviews) (Socher et al., 2013), SenTube (automobiles and tablets), (Uryupina et al., 2014), or the Thelwall Dataset (microblogs), (Thelwall et al., 2010).

Little work has been devoted to SA within the scope of migration. Shalunts and Backfried (2016b) investigate sentiment expressed in traditional media collected during the Refugee Crisis in Europe in English, German, Russian and Spanish. Backfried and Shalunts (2016) investigate shifts in sentiment around the events of New Year’s Eve in Cologne 2015 in English, German and Russian.

Even though the field has made considerable progress in the last decade, several challenges still remain to be solved and thoroughly addressed in the field of SA. This concerns a wider set of languages as well as linguistic phenomena such as: mixed-polarity expressions, the use of non-standard spellings, sarcasm and irony, negation, amplification, the of comparative language, shifters (words moving positive words towards more negative sentiment), reducers and the use of emoticons (Barnes et al., 2019).

5.1.3 SA Approaches

Approaches in SA can be divided in two broad categories: machine learning (ML) and lexicon-based ones. Historically, lexicon-based approaches preceded ML-based approaches with current approaches typically relying on neural-network-based-approaches, embeddings and transcoders (typically on pre-trained models which are adapted or extended in the process).

Lexicon-based approaches use a curated sentiment-lexicon, associating each entry with a specific sentiment and score. Units of processing may be on the word, sub-word (morpheme), or multi-word (idioms) level and phenomena such as negation, amplification, reduction etc. be accounted for. These approaches do not rely on labelled training data and exhibit robust performance within the domain they target. A prominent representative of this approach is SentiStrength (Thelwall et al., 2010).

Supervised ML-based approaches are implemented as binary (positive, negative) or n-ary (positive, negative, neutral) classifiers which are trained in labelled data. The dependency on such labelled datasets can be considered a major drawback as in many cases, such datasets do not exist and creation is costly in terms of resources and sometimes even outright impossible.

Current state-of-the-art models typically rely on features extracted in an unsupervised manner, mainly through existing, pre-trained embedding models (Mikolov et al., 2013). Within this setting, words are represented as a function of their contexts which allows ML algorithms to generalize over words (or tokens) occurring in similar contexts and with similar representations.

As in numerous other fields of NLP, the scope of systems (e.g. the set and number of different categories to be detected) as well as the datasets used for evaluation differ widely according to the targeted language, domain and whether SA is applied to content from traditional or social media sources.
Barnes et al. (2017) provide an overview of the performance of state-of-the-art systems on a set of state-of-the-art datasets. Their analysis on a set of six benchmark datasets shows that bidirectional LSTMs perform well on all datasets evaluated in general. LSTM and biLSTM seem to be particularly good on fine-grained tasks and embeddings trained jointly for semantics and sentiment perform well on datasets which are similar to the respective training data. A standard baseline system based on L2-regularized logistic regression on a bag-of-words representation likewise provides excellent results, outperforming more elaborate setups on several test-sets.

### 5.1.4 Performance of SOA

As outlined above, SA systems are typically measured in terms of accuracy. Taking into account inter-annotator-agreement allows to compare SA performance with typical human performance. The latter appears to be particularly of relevance for more complicated domains (where cultural, political, societal, etc factors play an important role) and when sentiment is not limited only to binary positive/negative granularity but to a richer and more fine-grained set of categories.

Wang and Manning (2012) provide performance figures on 8 datasets using a set of linear-classifiers and find that the use of bigrams consistently improves performance. In a recent experiment, Barnes et al., (2017) evaluate 5 state-of-the-art methods and two baseline methods (one of which is the standard baseline mentioned above) on six standard test-sets. The evaluation is performed for English data only and yields (macro-averaged23) accuracies of between 45.6% and 83.1%. Performance varies dramatically between test-sets with Stanford Sentiment Treebank (SST-fine, movie reviews with 5 levels of sentiment) scoring lowest and SST-binary (same dataset but with binary classification only) the highest.

As the field of SA is a very active one, the state-of-the-art on the various sub-fields is constantly being pushed by new developments. The website [http://nlpprogress.com/english/sentiment_analysis.html](http://nlpprogress.com/english/sentiment_analysis.html) provides an up-to-date overview of the latest developments, performance numbers, links to papers and datasets.

Due to the scarcity of work in the area of SA and migration-related topics and a lack of corresponding resources, no formal performance of SA has been published on migration-related data.

Both, Backfried and Shalunts (2016) and Shalunts and Backfried (2016b) examine the sentiment expressed in traditional and social media on migration-related topic (the refugee crisis) and note that shifts in sentiment can be aligned with notable real-world events involving refugees, that sentiment in traditional sources within one country (one language) frequently coincide and that SA can be employed to detect individual sources which publish extremely negative news (with an obvious agenda of doing so).

### 5.2 Relevance to and approach within MIRROR

The MIRROR project overall aims at detecting (potential) misperceptions about Europe and Europeans in the eyes of potential migrants and citizens who live abroad. Such misperceptions have the potential to lead to friction and eventually – if not handled or countered properly – may lead to threats. A fundamental task in

---

23 computing measures over all classes together rather than individually and then averaging the resulting values
the detection of misperceptions is to detect sentiment and polarity in media covering migration and migration-related issues. SA will be employed as one source of information for this process.

As the setting of MIRROR is clearly an international and multilingual one, any technology involved should allow to support several relevant languages and adaptation to the domain of migration and any sub-domains related to MRSCs.

Based on prior work of SAIL LABS (Shalunts and Backfried, 2015), the SentiSAIL system will be used for the SA of content all languages available to MIRROR via the Media Mining System. SentiSAIL is based on SentiStrength (Thelwall et al., 2010), a lexicon-based approach but extends this with several features (e.g. for morphology, negation, boosting (amplifiers and reducers), idioms, emoticons) and with models for several languages (both, improving performance for languages for which SentiStrength provides models as well as proving models for further languages). Furthermore, acknowledging that frequently different types of sentiment may be expressed jointly, SentiSAIL implements a four-way classification into negative, positive, mixed and neutral classes. Mixed classification designates the presence of both, negative as well as positive elements in a statement. On test-sets for English, Russian and German content, SentiSAIL achieves an accuracy of 75.3%, 79.3% and 90% respectively (Shalunts and Backfried, 2015).

**Progress made within MIRROR**

- Extension of SA patterns to migration-specific ones (e.g. Krimmigration (German neologism formed from the words kriminell and Migration – criminal migration))
- Creation of initial SA models for Urdu and Pashto

**5.3 Component Integration**

The SA component will receive text as input (payload of a REST request) and provide a list of annotations of detected sentiment (polarity: positive, negative, neutral or mixed and intensity as float values) as output.

**Input:** “Migration has caused a lot of social unrest and protests in Turkey.”

**Output:**
```
{
  "response" : {
    "type" : "annotations",
    "annotations": null,
    "features": {
      "polarity": "NEGATIVE",
      "positiveSentiment": "0.0",
      "negativeSentiment": "-2.2"
    }
  }
}
```

**5.4 Models/Technologies/Results/Insights/Next Steps**

The Sentiment Analysis component of MIRROR is built upon SentiSAIL currently employed by SAIL LABS. It provides polarity as well as intensity of an input statement. The component currently supports 16 languages. Among these set, several language spoken in countries of origin (COO), countries of transition (COT) and
countries of destination (COD) of migrational movements can be found, such as Urdu, Farsi, Pashto, Arabic, Turkish, Greek, German, Italian or Spanish.

Whereas the rule- and pattern-based nature of the current approach puts it at a disadvantage compared to more recent state-of-the-art systems, it also allows for rapid extension, experimentation, adaptation to migration-relevant domains and topics as well as to further languages should the need arise. Work so far has thus concentrated on the (multilingual) extension of existing models to concepts encountered during work on MRSCs. This allows to enrich the continuous stream of content processed by the Media Mining System and imported into the MIRROR framework (as soon as it becomes operational) as well as to classify textual input to the MIRROR system. Extensions have been carried out for Urdu and Pashto, leading to initial systems for SA for content in these languages. We envision to extend the set of patterns in a semi-supervised manner by clustering documents according to existing patterns and detection of elements and expressions which co-occur frequently with these existing entries. Furthermore, due to the fact that SentiSAIL labels sentences, paragraphs and complete documents with polarity and intensity, we plan to produce (migration-related) datasets which can be used for the training of state-of-the-art ML-based systems.

6 Document Clustering

This task focuses on mining and tracking topics discussed among migrants across media. Our goal is to examine, extend, and adapt advanced methods for topic detection and temporal topic modeling. We are working on the development of novel methods for deriving topics from multi-modal and heterogeneous data sources. In conjunction with WP6 we are also investigating joint modeling of topics and the networks/communities that could form around them. These methods do not only leverage advantages of different types of data to improve the modeling accuracy but also provide interpretable ways to characterize communities based on their associating topics. The detected topics will be exploited also in WP5, by the developed multi-modal approach for the summarization of media collections.

The targeted topics to be identified in this task are guided by the collection and taxonomy of Migration Related Semantic Concepts (MRSC) devised in WP2. However, the ongoing subtask of putting together a comprehensive dataset of annotated documents that match our MRSC is a challenging and time consuming endeavor. Notwithstanding the foregoing, we have achieved significant progress in the study and adoption of topic modeling approaches for MIRROR. At first, we are using either already existing datasets or creating our own general purpose, still relevant, annotated collection. Later, the developed models will be refined and validated against the MRSC-topics specific task.

Currently, we are following two complementary approaches: supervised and unsupervised topic detection. The first one will allow us to classify documents into known and well-defined topics (i.e., MRSC). Meanwhile, the unsupervised models will help us discover new subtopics and track evolution of different subjects.

6.1 Related Work

Existing topic modeling methods can be categorized into two approaches: (i) matrix factorization and (ii) probabilistic generative models. Notable methods of the former approach are Latent Semantic Analysis (LSA) (Deerwester et al., 1990) and Non-negative Matrix Factorization (Xu et al., 2003), and the pioneering ones
for the latter are Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003). Later, there have been numerous works on adapting LDA for short texts. There are two main approaches in this adaptation: (a) limiting the number of topics per documents, e.g., (Zhao et al., 2011), and (b) enriching the documents, e.g. (Yan et al., 2013). All these models are however computationally expensive.

Recently, there are works that propose to leverage advanced computing devices (e.g., the GPUs) for accelerating LDA-like models (Miao et al., 2017, Lin et al., 2019). Some others propose to exploit pre-trained word embedding corpora for incorporating the background knowledge about words (Das et al., 2015, Li et al., 2016). There are also few works that propose to replace the bag-of-words representation by the sequence of words and employ a hybrid of topic and language model for modeling the sequences (Dieng et al., 2016). However, the performance of these models has not been comprehensively examined. For example, the documents’ topic distributions learned by these models are not evaluated and reported in their corresponding papers.

As presented here, with the large size of data streams, there are several works from different approaches that propose automatic methods for this task. However, there are only a few small benchmark datasets that are publicly available for evaluating the proposed methods. The lack of large datasets with fine-grained ground-truth implicitly restrains the development of more advanced techniques, particularly for supervised modeling. Semi-supervised approaches Conrad and Bender (2016) try to get around this problem by using top-level clusters for news events based on an editorially supplied topical label, known as a ‘slugline’.

### 6.2 Unsupervised Neural Topic Modeling

#### 6.2.1 Problem statement

Unsupervised topic modeling is a powerful technique for uncovering latent topics from large text corpus (Boyd-Graber et al., 2017). In principle, this includes a wide range of methods for co-clustering of documents and words appearing in these documents. Due to its high performance and interpretability, topic modeling techniques have been applied in many important applications and attracted much interest from researchers (Blei, 2012). Despite a rich literature, traditional topic models are computationally expensive and thus do not cope well with very large-scale datasets. Moreover, as these models rely on the bag-of-words representation of documents and co-occurrences of words in the documents, they suffer from sparsity problems, particularly when applied to short texts. Also, by using words as atomic units, they do not capture well the general background of the words, e.g., the semantic relatedness among them. Last but not least, as the bag-of-words representation ignores the order among the words in the same documents, these models fail to capture other important aspects expressed in the documents, e.g. sentiments and stances toward the topics discussed. For example, the two sentences “I like Android but I hate the iPhone” and “I like the iPhone but I hate Android” have the same bag-of-words representation, and hence traditional topic models cannot differentiate between them in terms of sentiments expressed toward “Android” and “iPhone”.

In this work, we therefore would like to address the above shortcoming in the existing topic models. Specifically, we aim to develop novel topic models that allow us to leverage background knowledge, both general and domain specific, about the words for dealing with the sparsity problem in short texts. We would
also like the models to be able to capture the sentiments expressed in the texts while being scalable to large datasets.

6.2.2 Mirror Approach

In this work, we would like to fill the gap in literature on topic modeling for short texts as discussed above by (i) first examining the performance of the recently advanced models, and then (ii) developing novel ones that are able to incorporate background knowledge about words to uncover both topics and sentiments expressed toward the topics.

To do so, we will first collect comprehensive large datasets with ground-truth labels for documents. These datasets would allow us to evaluate the performance of the models comprehensively by examining the learned topics and documents’ topic distribution in downstream tasks, e.g., document classification and clustering.

Later, based on the above evaluation, we would determine the approach for developing the novel methods. Generally, we expect that a deep generative probabilistic model would help to improve the existing models as they can leverage both the high performance of deep neural networks while enjoying the high interpretability of probabilistic models.

6.2.3 The Datasets and Next Steps

We have collected the following datasets for conducting experiments in this work:

- **Twitter**: this dataset consists of 200K+ tweets published by Twitter accounts of big news agencies that are dedicated to one of 11 news categories, e.g., CNN Politics, and BBC Weather. For each account, we use its category to determine the label of its tweets. That is, all tweets published by CNN Politics are assigned with the Politics label, and so on.

- **Headlines**: this dataset consists of 105K+ titles of news for top 30 events in W2E dataset (Hoang et al., 2018b)

- **Paper titles**: this dataset consists of 120K+ titles of preprint papers submitted to the Computer Science category of arXiv - the free distribution service and an open-access archive hosted by Cornell University. The labels of each paper are the fields that the paper belongs to, e.g., Artificial Intelligence, and Databases, etc.

As shown above, the collected datasets are large and diverse in nature, allowing us to evaluate the models more comprehensively.

Our next step is to start developing a deep generative probabilistic model. We will identify and implement some state of the art algorithms to use as benchmark over the collected datasets.
6.3 **Supervised Deep Topic Modeling with Personalized Attention for News Articles**

6.3.1 **Problem statement**

Topic models have been mainly exploited to detect, track, and describe topics from a stream of broadcast news reports. The application of these techniques ranges from mitigating bias by presenting multiple points of view for the same subject (Park et al., 2012), comparing information received by different audiences (McKeown and Evans, 2005), to simple topic summarization with data from various sources (Yang et al., 2019). However, as also described before, most existing topic models neglect semantic information and structural features derived from standard journalistic practices. Moreover, rather than on the topic, many of the studies are centered on the encoding of the documents to allow a later similarity analysis (Blokh and Alexandrov, 2017).

Naturally, many of the objectives in this study (e.g., leverage of background knowledge) are aligned with the research presented in the previous section (see Section 6.2). But, in particular, we want to exploit the predefined structure of our MRSCs to develop supervised models that are topic-centered.

6.3.2 **Mirror Approach**

We plan to take advantage of recent developments in deep learning models to extend the existing approaches to the topic modeling problem. First, under the idea of the inverted information pyramid structure followed by trained journalists when writing news articles, we use the title and first paragraph as representative parts of an article. Second, we assign a weight to each word within each article depending not only on its frequency but also its relative positions (e.g., words appearing in the title are given a higher weight).

Through a deep neural network and word embedding, we calculate an article representation that is based on the title and first paragraph of each article. We use a set of convolutional layers to extract contextual information from the content. Additionally, for each document, we calculate a weight vector representation that captures syntactic properties and word importance within each article. These weights are used to create a word-level personalized attention network. For the topic encoder module, news-level personalized attention is used to build informative topic representations. Finally, with a larger dataset of labeled articles, we train, validate, and test our model.

6.3.3 **The Datasets and Next Steps**

Currently, we are conducting our experiments over a dataset consisting of 200K+ news articles classified in more than 3000 topics (Hoang et al., 2018b). In the future, we will experiment with other datasets, including MRSC-topics.

For the model, we will experiment with different architectures (e.g., estimating the contribution of batch normalization, an optimal number of convolutional layers, RNN, etc.). Furthermore, we plan to conduct features analysis such as Layerwise Relevance Propagation (Bach et al., 2015) for determining the relevance.
of each feature in the outcome of the model. This can help, for example, to characterize topics in terms of the most important words or most representative articles.

7 Detection of Computer-generated Content

Much disinformation is present in the modern media landscape. This is true also for MIRROR-related topics such as migration, which easily become the target for polarized views. In this context, it becomes important to be able to analyze the textual content of editorial information related to migration, such as news articles and user comments on such articles. The news articles are of high importance as they are often linked to, and spread heavily, through social media, since news articles are often considered to be more trustworthy than other content. User comments, on the other hand, can have an influence on our understanding of the “average citizen’s” opinion about a topic.

Already today, there is a growing concern of fake news articles being created, with purposes such as generating clicks and driving traffic to a site to generate ad revenues. Similarly, user comments on news articles often have to be heavily moderated due to a toxic mixture of everything from hate speech to nation state-based propaganda trolls attempting to influence public opinion on specific topics. The advent of powerful deep learning-based generative techniques has led to the ability to generate texts, synthesize high-resolution images of people who does not exist, and create deep fake videos in which voices and faces are manipulated with quickly increasing trustworthiness. Therefore, there is a concern for automated creation of fake news articles and “user-generated” comments. This calls for the development of new algorithms and tools able to distinguish between texts that have been automatically created, and those that have not.

7.1 Automatic Text Generation

With the advent of deep learning-based text generation models, the quality of computer-generated text has been greatly improved. A few years ago, character and word level RNNs\(^\text{24}\) started to produce diverse text that looked syntactically correct, but on closer inspection hardly could be mistaken for real human-generated text due to its lack of coherence and semantically relevant sentences. However, this situation is changing quickly, not least due to the introduction of large-scale neural models consisting of billions of parameters, trained on huge amounts of available text resources.

In the below literature review the focus is on the Transformer-based models GPT-2 (Radford et al., 2019), GROVER (Zellers et al., 2019), and CTRL (Keskar et al., 2019). Even larger models such as NVIDIA’s Megatron-LM (Shoeybi et al., 2019) continue to be published, but share most of the characteristics of the models presented here.

\(^{24}\)http://karpathy.github.io/2015/05/21/rnn-effectiveness/
7.1.1 GPT-2

GPT-2 (Radford et al., 2019) is based on a Transformer-model originally developed for machine translation (Vaswani et al., 2017), but unlike the original encoder-decoder architecture, GPT-2 only consists of the decoder part of a Transformer, as this component is everything that is required for text generation. Without going into too much technical details, GPT-2 consists of a large number of decoder blocks stacked on top of each other, which in each time step predict what the next token should be, given the current tokens being generated or given as input. Each decoder block consists of a layer of masked attention followed by a small feed-forward neural network, which applies a non-linear transformation of the current text representation. The more layers the model has, the longer it takes to train and the more training data is required, but the most powerful GPT-2 model released by OpenAI has been trained on a diverse dataset of 8 million web pages with a simple objective very relevant for text generation: to predict the next token (word) given all of the previous tokens (words) within some text. Once it has been trained, new texts can be sampled from the model.

The original GPT-2 model makes use of top-k truncated sampling. GPT-2 can generate synthetic texts unconditionally, but more interestingly it can be primed with relevant input and generate a continuation of a given text sequence. The model adapts to the content as well as the style of the text to be conditioned on, which allows for some control of what is being generated by the model. The text generations from GPT-2 can be made more relevant for a specific domain by fine-tuning the model, i.e., a pre-trained model can be trained further on domain-specific data. For example, by fine-tuning a model on Twitter data related to migration, it becomes better at generating migration-related tweets.

In a study (Solaiman, 2019), Amazon Mechanical Turk was used to compare aggregate credibility scores (on a scale from 1 to 10) assigned to news stories generated by different sizes of GPT-2 models. The largest 1.5B parameter model got an average score of 6.91, which was higher, but not significantly so, than the 774M model.

7.1.2 GROVER

Similar to GPT-2, GROVER (Zellers et al., 2019) is also a large-scale Transformer-based model, but unlike GPT-2 it is in addition to the text body of an article also taking metadata such as domain, date, author, and headline into account when generating new texts. This allows for more fine-grained control as it, e.g., can be used to generate a news article conditioned on the style of a particular New York Times columnist, or on a specific topic (as being probed by given a particular title as input). As sampling strategy GROVER is using nucleus sampling, i.e., for a given threshold $p$ the model is in each timestamp sampling from the top-$p$% of the entire vocabulary, using a cumulative probability distribution. In their paper, the authors present a study in which it is shown that workers on Amazon Mechanical Turk assign an overall trustworthiness score that is higher for propaganda articles generated by GROVER than for human-written articles from known propaganda web sites.
7.1.3 CTRL

CTRL (Keskar et al., 2019) is a 1.6 billion parameter Transformer-based language model. The main difference to GPT-2 is that it is trained with over fifty different control codes which allow for better control of what is being generated, letting the user explicitly control things such as style, genre, and specific entities. This allows for things such as generating negative reviews of a specific product. CTRL uses a greedy sampling strategy with penalized sampling to avoid producing repetitive text.

7.1.4 PPLM

Fine-tuning pre-trained language models is one way to allow for better control of what is being generated, but doing so requires much hardware resources and data. PPLM (Dathathri et al., 2019) allows for combining language models with small attribute models which may be 100,000 times smaller than the pre-trained language model and still allow for effective steering of the generated output. The attribute models can be simple bag-of-words models that are used to represent the topics of interest. As an alternative, attributes such as positive or negative sentiment can be expressed using small and shallow discriminative models trained on a dataset labeled with the desired target class.

7.2 Detection of Computer-generated Text

Given the recent developments in neural text generation it is clear that it is becoming increasingly difficult for human readers to tell computer-generated texts apart from news articles and comments created by humans. This situation calls for algorithms and tools able to assist humans in detecting generated text. Unfortunately, however, such detection is currently not described much in the literature. Hovy (2016) made an early attempt to detect statistically-generated fake reviews generated by an n-gram language model. For detection they use n-gram features as input to a logistic regression model trained on a dataset containing true and fake reviews. A similar model has been suggested by OpenAI as a baseline system for detecting text generated by GPT-2 models, available from their GitHub repository. The GROVER system has been shown to detect its own GROVER-written fake news with high accuracy, and in follow-up research published on their blog the system is also shown to detect fake news articles generated by other language models such as GPT-2, where initial results suggest accuracies around 95% given a large training dataset with representative examples. Finally, the visual open source tool GLTR (Gehrmann et al. 2019) is intended to support human detection of generated text by highlighting various statistical information related to the text, which relies on that most text generation algorithms make use of biased sampling techniques which impact the generated text.

25 https://github.com/openai/gpt-2-output-dataset/

26 https://grover.allenai.org/detect
### 7.3 Component Integration

The component will receive a title and a body of text as input (payload of a REST request) and provide a score corresponding to the likelihood of the text having been created automatically.

**Input:** “Israel says Palestinian refugees number in the thousands, not millions as some have claimed.”

**Output:**

```
{
    "automation_score": "0.876"
}
```

### 7.4 Ongoing Technical Work

The approach taken to detect computer-generated migration-related text comprise of the following ongoing technical tasks:

1. Collect representative negative training examples for real migration-related news articles and user comments on such articles.
2. Generate representative positive training examples of computer-generated migration-related news articles and comments using various state of the art neural language models.
3. Train machine learning-based classifiers able to discriminate between real and computer-generated migration-related news articles and user comments.
4. Incorporate the developed classifiers into a prototype tool helping MIRROR users to detect migration-related texts generated by language models.

### 8 Summary and Next Steps

In this deliverable, the first set of components and models for text-analysis and -enrichment, developed and extended within WP4 and their background have been presented. The diversity of technologies targeted by WP4 is quite extensive, with the respective state-of-the-art and level of work differing accordingly. Several fields are very active with regard to research which warranted to thorough review and update of the state-of-the-art. The methodologies and technologies used have been covered to contextualize the ongoing work, initial results and future plans. Several of the components have reached a level of maturity to be ready for packaging as an initial set of components within the MIRROR system. While further investigation will continue during project year two, the deployment of use of models will provide the basis for end-user feedback and refinement.

At the outset (as described in the DoW), the overall approach for WP4 was described as:

*The WP will take particular care to: reuse existing models wherever possible and build upon existing infrastructure, to develop and extend models in a flexible way to allow for swift adaptation to further languages/dialects/content – as the domain of migration is a very dynamic one expected to change even during the duration of the project, and to apply models and technologies which can be duplicated for further languages with minimal effort (ie explicitly prefer light-weight models over complex ones).*
Following this approach, pre-existing models and technologies have been employed to kick-off processing and data collection from the project start. This approach is especially important as many of the technologies are machine-learning-based and as such require data for training and evaluation.

However, substantial progress was made regarding data-ingestion as well as enrichment (as outlined in the chapters above).

The following items provide an overview of these achievements and activities:

- Assignment of origin-information to more than 7000 sources to be able to localize information
- Extension of NE categories and addition of NEs for migration-related topics
- Creation of an initial component for SA in Urdu and Pashto
- Significant progress in the study and adoption of topic modeling approaches for MIRROR
- Study, evaluation and adoption of machine learning models that can learn to distinguish between human and computer-generated comments related to migration

On the integration side, the plan is to *dockerize* the components and integrate them into specific workflows within MIRROR. Furthermore, payloads and data formats will be defined according to the output of work on the information model in WP7.
9 Bibliography


Barnes, J., Klinger, R. and Schulte im Walde, S., Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment Datasets, Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, p 2-12, Copenhagen, Denmark, September 2017.


Conrad, J. G. and Bender, M., Semi-supervised events clustering in news retrieval. In NewsIR@ ECIR, pages 21–26., 2016


Kohlschütter, Ch., Fankhauser, P. and Nejdl, W., Boilerplate Detection Using Shallow Text Features, WSDM 2010 - Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, pages 441-450, 2010


McKeown, K. and Evans, D. K., Identifying similarities and differences across english and arabic news., 2005


Shalunts, G., Backfried, G., SentiSAIL: Sentiment Analysis in English, German and Russian, International Conference on Machine Learning and Data Mining, Hamburg, Germany, 11-24 July 2015


Shalunts, G., Backfried, G., Multilingual Sentiment Analysis on Data of the Refugee Crisis in Europe, 5th International Conference on Data Analytics, IARIA 2016, Venice, Italy, Oct 9-13, 2016 (2016b)


Wang, S., Manning, Ch., Baselines and Bigrams: Simple, Good Sentiment and Topic Classification, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, Korea, 2012


Yang, P., Li, W., and Zhao, G., Language model-driven topic clustering and summarization for news articles. IEEE Access, 7:185506–185519, 2019


Zheng, A. and Casari, A, Feature Engineering for Machine Learning, O’Reilly, 2018