A multi-lingually applicable journalist toolset for the big-data era

G. Kiomourtzis and **G. Giannakopoulos** and **V. Karkaletsis**
NCSR "Demokritos", Athens, Greece
{ggianna|gkiom|vangelis}@iit.demokritos.gr

A. Kosmopoulos
SciFY PNPC, Athens, Greece
akosmo@scify.org

Abstract

This paper overviews the NewSum Toolkit toolset, providing a full set of Natural Language Processing tools aimed to support journalists and publishers during the news writing phase. The toolset contains a set of services, currently integrated in commercial products, that can overview and summarize thousands of sources (social media and news feeds) in different languages, support automatic article classification based on existing or new taxonomies, and facilitate slug-line generation. The overall system builds on well-established, open technologies and tools to provide a backbone that can empower any publishing platform.

1 Introduction and Related work

Several advances in sub-domains of Natural Language Processing (NLP) have, for several years, been overlooked by the media industry, because oftentimes the efficiency of the tools was insufficient to be applied in a real-world, open domain setting. Recent years have, however, redefined the relation between NLP and the world of media. The redefinition is related to both the changes in the mass media landscape [Curran, 2010] — leading to data journalism, crowdsourced [Brabham, 2012] and crowdfunded journalism [Aitamurto, 2011] — but also the updated focus of scientific efforts and results towards real world and industrial problems: multi-document news summarization [Dang and Owczarzak, 2008; Giannakopoulos *et al.*, 2011], argument summarization [Swanson *et al.*, 2015], forum and socal media summarization [Kabadjov *et al.*, 2015] to scientific summarization [Qazvinian *et al.*, 2013].

Research has led a number of efforts to empower news summarization, with the Document Understanding and Text Analysis Conferences (e.g. [Dang, 2006; Dang and Owczarzak, 2008]) and with NTCIR[1]. Throughout the years, many different approaches on summarization have been proposed, ranging from seminal works on simple, frequency or keyword based schemas [Luhn, 1958] to latent semantic spaces and network-based methods [Mihalcea, 2005]. However little has been done to empower the journalist. Even though there exist several online tools related to summarization, most focus on the reader-consumer or to automatic summarization per se. Such tools include search result summaries (e.g. JistWeb and Ultimate Search Assistant), and single document summarizers (TLDR Reader, ReadBorg).

In this work we overview NewSum Toolkit, which comes to address this exact need: bringing reusable, state-of-the-art Natural Language Processing to the journalist, empowering news writing and publishing. We then conclude, providing a glimpse of future extensions to the toolkit.

2 NewSum Toolkit: An overview

The NewSum Toolkit toolset can be broken down into a number of services that cover a variety of journalistic needs, as follows.

Data gathering This service provides a number of web services that allow management and monitoring of media sources. These sources can be news feeds (e.g. RSS/Atom feeds), blogs and websites or social media sources (e.g. Twitter, Facebook). This service allows journalists to follow all their sources of interest under a unified view.

Summarization The summarization component of the system, implements two critical functions. First, it identifies events, as these are reported across sources. To this end, a number of NLP methods are combined to first measure the similarity and then group content items (e.g. posts, articles) into events. The second function this component achieves is that of summarization. Essentially, employing language-agnostic summarization methods based on open n-gram graph technologies [Giannakopoulos *et al.*, 2014], the system undertakes the task to provide a snippet-based (extractive) summary of each event. This summary aims to provide representative information, while removing the expected (and often extreme) redundancy from the varying sources. The system ascertains that the summary contains all the links back to the original sources, allowing verification and minimizing error.

Trend detection In the trend detection component, events are followed throughout their life-span to detect importance over time. Using varying windows of time, an importance index is calculated, which represents overall

[1] See http://research.nii.ac.jp/ntcir/ for more information.

and instantaneous trend. The output of this component allow for the ranking of incoming events and can help the journalists prioritize their focus.

Automatic Classification Within the news generation process, there exists a number of near-trivial tasks that can take significant time, such as the annotation of news with specific classification codes (e.g. based on the International Press Telecommunications Council or IPTC media topics classification[2]). To help journalists, the automatic classification component can suggest categories and annotations, minimizing human effort. This component employs machine learning techniques to improve over time, based on the actual annotations and corrections of the users.

Slugline generation Sluglines are another type of repetitive and time-consuming task that NewSum Toolkit comes to support: a slugline contains a few keywords and phrases, aiming to outline the setting of an event. Based on named entity recognition and keyword extraction methods, the system undertakes the task of suggesting a slugline, which can then be finalized by the journalist.

Overall, the above set of tools have been created to facilitate the whole life-cycle of news generation, from the identification of important (or potentially important) events, to the actual writing of the article. The system is built to be applicable over a large range of languages, with minimal fine-tuning, since it relies on mostly language-agnostic approaches. Furthermore, each component has been implemented with a parallel execution approach and easy cloud-based integration, making the system "big-data"-enabled and allowing scaling to tens of thousands of sources with minimum effort.

In the following paragraphs we elaborate on the NewSum Toolkit components, providing some insights on the related technical approach and provided outputs.

2.1 Data gathering

The data gathering service supports RSS/Atom feeds, blogs, web pages scraping (through customization), and different kinds of social media, i.e. Twitter, Facebook, etc. The service is designed in an easily extensible way, to support other kinds of sources with ease. The news/blogs/web sites module operates regularly and updates the database with new entries, avoiding duplication, while the social media modules are targeted on a supplied — by the journalist — group of accounts, which are monitored in short time intervals.

The module is known to operate well on an 8-CPU core server with 16GB of RAM, in a setup of more than 1000 sources (including 200 monitored social media accounts) with a 15 minute refresh interval. The news module uses a distributed NoSQL solution (mongoDB, cf. [Abramova and Bernardino, 2013]) as a persistence mechanism, while the social media data are temporarily stored in a relational database — due to a more relational structure of the content. Further processing unifies the metadata in mongoDB to improve scalability.

[2]See http://cv.iptc.org/newscodes/ for more information regarding IPTC codes.

The RSS/blog/web sites module is built using open source Java libraries, covering RSS feed parsing and HTML page (DOM) parsing. For the social media data gathering, NewSum Toolkit utilizes the corresponding most popular open source Java libraries for each respective medium (Twitter, Facebook, etc.), which build upon the corresponding REST APIs (e.g. Twitter API, Facebook Graph API).

Once again, the data gathering is a highly parallelizable process, which allows for scalability in both storage and execution.

2.2 Summarization

Summarization in NewSum Toolkit is a three stage process. In the first stage, documents are grouped into clusters that we expect will refer to one topic or event. In the second stage, the systems processes each cluster, determining "sub-topics" — i.e. aspect of the topic/event discussed in the cluster documents. Then, each subtopic is represented by a set of sentences that maximally cover the subtopic, with minimal redundancy.

The clustering process can be broken down into the following individual steps: meta-data-based document grouping; similarity calculation within groups; determination of clusters. The steps are elaborated below.

Initially, documents are mapped to groups according to their meta-data, i.e. items from the same news category, etc. Essentially, this is a blocking step to reduce the complexity of pairwise comparisons between documents and allow efficient, large scale analysis. This approach also covers a user need which connects specific sources to be appropriate for specific news categories (e.g. foreign policy news vs. music industry news). The meta-data assigned to each source are provided during the setup phase of the NewSum Toolkit. Other meta-data may originate from the item itself (e.g. category specification in the NewsML format of a news item).

For each item pair, specific similarity measurements are extracted, which are then combined with heuristic rules to determine pairs of related documents. The similarity measurements use pre-processing and text analysis to extract a number of features, related to named entities and morphological characteristics of the text.

Given the results of this pairwise matching, the system determines the transitive closure of the matching relation to define clusters. In other words, a graph is generated between documents, where edges depict a "refer-to-the-same-event" relation between documents. Whenever there exists a path between two documents in this graph, the documents belong to the same cluster. Thus, the result of the clustering is a hard (i.e. non-overlapping) set of clusters, where each cluster of documents is considered to represent documents referring to the same event (cf. Figure 1).

For each topic/event cluster, we then identify what we term "sub-topics". A sub-topic is a cluster of sentences that is expected to reflect a specific view of a topic (e.g. the financial vs. the political aspect). To determine these sub-topics, we first represent sentences as character n-gram graphs [Giannakopoulos *et al.*, 2008]. We then compare all pairs of sentences, based on their character n-gram representation. We use character 3-grams and a neighborhood distance of 3 to

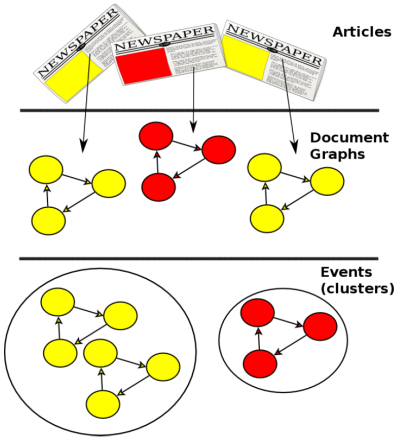

Figure 1: Clustering for event detection

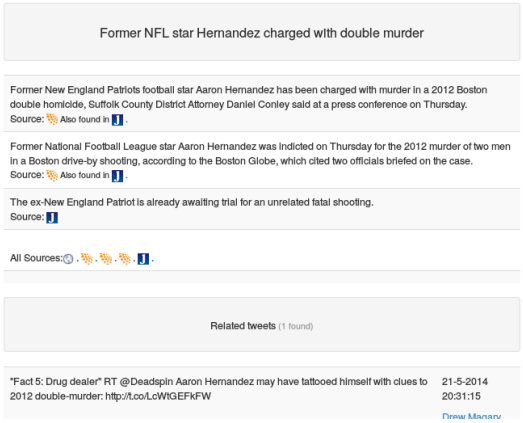

Figure 2: A cross-media news summary

represent the sentences. These values have been shown to perform well in a variety of settings. The output of this step is a similarity matrix between sentences, based on the Normalized Value Similarity (NVS) between the n-gram graphs of the sentence pairs [Giannakopoulos *et al.*, 2008].

We then apply Markov Clustering (MCL) [Dongen, 2000] on the similarity matrix. The result of this process is a set of hard clusters, identifying sub-topics. The summarization process concludes by selecting sentences that are most representative for the sub-topic, while limiting redundancy. To achieve this double goal, we first create representative "centroid" graphs per topic. We then form n-gram graphs for every candidate sentence. Then we sort candidate sentences based on the similarity of their graph to the representative graph of the sub-topic. Then, we run through the sorted candidate list, removing sentences that appear too similar (i.e. surpass a similarity threshold between each other).

The above method has been used in the MultiLing, multilingual multi-document, summarization challenge [Giannakopoulos *et al.*, 2015] with promising results (ranking in the top 50% of systems in the overall ranking across all languages) without fine-tuning. In the industrial system we apply fine-tuning to increase the performance on target languages.

We note that the system also holds the potential to link clusters between the news (RSS feeds) and the social media (e.g. Twitter). To this end, we employ n-gram graph similarity between the news and the social media clusters; if the similarity exceeds a heuristically-defined threshold, we connect the clusters as referring to the same topic. We illustrate such an example in Figure 2.

2.3 Trend detection

In order for the suite to detect trending events, it groups the news clusters (i.e. topics) into an abstraction defined as "story". Essentially, a story depicts the way topics are evolving in a time period. The significance of a specific event is closely related to the news sources that write about this event, i.e. the topic size. The trend detection algorithm operates in two ways: first it extracts the number of sources that determine a story in the time window that the story exists, and secondly it looks for instantaneous trend by only checking the topic size of the two most recent topics of the story. These two approaches are meant to indicate two different types of trends: instantaneous trend, which implies topic "hotness", but also overall trend, which implies robustness in the significance of topic. Essentially, the trend if the delta of the number of sources either across two individual time-points (instantaneous trend) or over a span of time (e.g. two days).

The above approaches allow the journalist to shift the focus from *potentially* important topics with reduced coverage — i.e. topics with significant instantaneous trend — to topics that persistently remain important over time, as appropriate.

2.4 Automatic classification

The goal of the automatic classification component is to suggest IPTC codes (i.e. topic-related codes from a hierarchy of formalized classes) for each news article. Our system uses one binary classifier for each class (IPTC code). During the prediction step a probability is computed for each class in order to suggest the IPTC codes with the highest probability to the journalist.

Annotated NewsML[3] articles are used for training the classifiers for each class. We first extract the text from the *NewsLineText* and *DataContent* tags of the NewsML file. We then transform the extracted text to feature vectors using a bag of words approach. Stemming procedures and TF-IDF transformation are also used. Adapted strategies regarding these prepossessing techniques take advantage of language-specific resources to support several languages (e.g.English and Greek in our current installations), but we do not elaborate on these fine-tuning processes here for lack of space.

An L2 Logistic Regression [Fan *et al.*, 2008] classifier is used to model each class. This method is quite accurate for binary classification, while also providing a probability for its prediction, which can be exploited for multi-label classification (i.e. by keeping highly confident predictions). Second it is quite scalable and since training and prediction for each

[3]See https://iptc.org/standards/ for more information regarding NewsML formats.

class is independent to the others, it can be run in parallel to improve scalability.

This process supports journalists in the annotation of their stories and minimizes the related effort.

2.5 Slugline generation

Slugline generation relies mostly on Named Entity Recognition (NER), to suggest a slugline to the journalist in the time of story writing and editing. We combine a set of entity lists (gazetteers) from a variety of sources (name/surname lists, organizations, etc.) with NER models (based on the OpenNLP toolkit [Baldridge, 2005]) to identify entities in the journalist article. Thus, the proposed slugline contains the identified set of entities. Ongoing work also builds upon keyword extraction techniques to supplement the extracted entities in the slugline and improve on our current approach.

2.6 User studies

We note that in the past [Giannakopoulos *et al.*, 2014] we had conducted user studies to identify the usefulness of the summarization, as well as the quality of the summaries.

To answer the question of whether our system (early user interface of the toolkit in 2013) facilitates news reading, we performed a small scale user experience experiment, limited to 18 Greek and 7 English beta testers [Giannakopoulos *et al.*, 2014]. These testers, who were recruited via an open call, were provided a questionnaire that measured different aspects of user experience. The question we will discuss here was expressed as "The use of NewSum allowed me to get informed of the latest news more globally and thoroughly than before". The answer allowed a 5-scale response from "I totally disagree" to "I totally agree". 11 out of 18 (61%) Greek users and 4 out of 7 (58%) English users have an answer of 4 or 5 to this question. Only 1 user per language thought that the system did not really help (2, in the 5-scaled response). The mean grade for Greek was 4 with a standard error of 0.24; for English the mean was 3.86 with a standard error of 0.46. Thus, these preliminary results indicate that users tend to believe that using the summaries can improve their news reading, fulfilling its purpose.

Two more user studies [Giannakopoulos *et al.*, 2014] on closed and open beta versions of the summarization system. The latter evaluation, which featured an improved version of the summarization system, returned 720 ratings mapped to individual users. 267 ratings were for English summaries and 453 for Greek, showing promising performance but also holding more findings. The overall rating distribution over all users and languages are shown in Figure 3.

The first finding of this study was that the language and the user are highly statistically significant factors for the results, and this was also shown by the average performances: for Greek the average was 4.14 (with a standard deviation of 1.07), while for English the average was 3.73 (with a standard deviation of 1.34). This showed that fine-tuning may make sense on individual languages and later versions of the system have this ability. In both languages the average performance was good, with more than 90% of the summaries having an acceptable (or better) grade for Greek and more than 80% for English. For a detailed description of the setting and findings, please consult the original work [Giannakopoulos *et al.*, 2014].

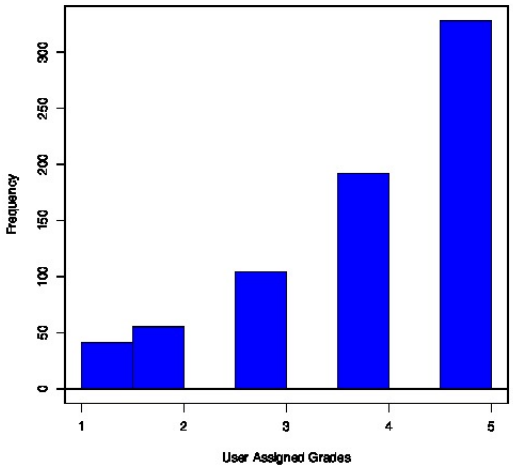

Figure 3: User study evaluation results over all languages and users

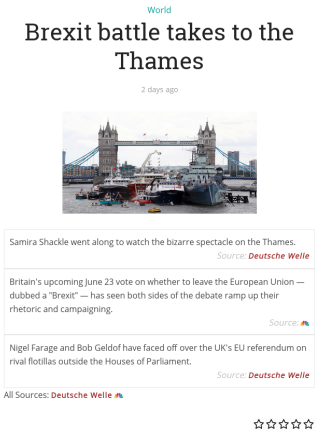

Figure 4: A summary snapshot of the web-based demonstrator of the NewSum Toolkit

In recent interfaces of the system, such as the one displayed in Figure 4, ratings are inherently supported to further improve the performance of the system over time (cf. Figure 4). A demonstrator can be found in the following URL: `http://newsumontheweb.org/`.

2.7 Challenges faced and lessons learnt

The combination of all the above technologies under a single product holds several challenges. We briefly describe the main challenges per module.

In data gathering, there exists a variety of embedded information even in RSS feeds. This information can vary from videos to advertisements. Oftentimes, the news feed itself is a flawed implementation of protocols, which provides partially malformed data. Thus, the integration of sources usu-

ally needs human effort to both clean-up and maintain over time.

In summarization, there exist domains where the clustering similarity threshold value varies significantly. In some cases, e.g. highly focused domains such as the car industry, the identifying elements are not topic-related terms, but rather named entities (car brandsor models). Thus, one needs to support several types of clustering algorithms, based on different features. We have also understood that cluster precision (being precise in what you put in the summary) is more important than recall (adding all related conent to the summary). The perceived value of the summary is significantly undermined if one sees irrelevant text (even a single piece) in the summary.

In classification, the training data offered per category heavily affect the performance of a classification — as expected in most classification settings. Thus, the user needs to be informed on best practices for the training of new categories. Oftentimes, returning a confidence per classification decision that the system took can further improve the usefulness of the system.

Finally, in slugline generation, fine-tuning is needed to achieve a balance between precision and recall. As an example, exhaustive name lists often include names that are also location names (e.g. Thessaloniki as a location and as a person name). Pragmatic knowledge biases our human view of the entity (Thessaloniki is almost always used as a location) and such knowledge needs to be integrated into the system to avoid misclassifications.

Overall, the automated algorithms form a robust basis, which however needs adjustments to provide production-level performance per domain of application.

3 Conclusion and future work

In this paper we overview the NewSum Toolkit toolset, aimed at empowering journalists through Natural Language Processing and Machine Learning tools. The toolset supports a number of stages in the story writing, from gathering information, to detecting important topics and writing the final text. NewSum Toolkit is essentially an infrastructure that can be integrated with any platform and tool through a friendly API, taking advantage of the cloud infrastructure and scalable, parallel algorithms to cope with the big data environment.

In the future plans we include a time-line view of news, supporting the journalist in the documentation of an event, by providing easy access to past knowledge. We also foresee a real-time editor module, empowered with automatic suggestions related to writing style, and with enrichment features that will use linked open data standards to annotate articles, to improve their publication efficiency.

Acknowledgments

This paper is supported by the project "Your Data Stories – YDS", which has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 645886.

References

[Abramova and Bernardino, 2013] Veronika Abramova and Jorge Bernardino. Nosql databases: Mongodb vs cassandra. In *Proceedings of the International C* Conference on Computer Science and Software Engineering*, pages 14–22. ACM, 2013.

[Aitamurto, 2011] Tanja Aitamurto. The impact of crowdfunding on journalism: Case study of spot. us, a platform for community-funded reporting. *Journalism practice*, 5(4):429–445, 2011.

[Baldridge, 2005] Jason Baldridge. The opennlp project. *URL: http://opennlp. apache. org/index. html,(accessed 2 February 2012)*, 2005.

[Brabham, 2012] Daren C Brabham. The myth of amateur crowds: A critical discourse analysis of crowdsourcing coverage. *Information, Communication & Society*, 15(3):394–410, 2012.

[Curran, 2010] James Curran. The future of journalism. *Journalism studies*, 11(4):464–476, 2010.

[Dang and Owczarzak, 2008] H. T Dang and K. Owczarzak. Overview of the tac 2008 update summarization task. In *TAC 2008 Workshop - Notebook papers and results*, page 1023, Nov 2008.

[Dang, 2006] H. T Dang. Overview of duc 2006. In *Proceedings of HLT-NAACL 2006*, 2006.

[Dongen, 2000] Stijn Dongen. Performance criteria for graph clustering and markov cluster experiments. 2000.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[Giannakopoulos *et al.*, 2008] George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5, 2008.

[Giannakopoulos *et al.*, 2011] George Giannakopoulos, Mahmoud El-Haj, Benoit Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma. Tac2011 multiling pilot overview. In *TAC 2011 Workshop*, 2011.

[Giannakopoulos *et al.*, 2014] George Giannakopoulos, George Kiomourtzis, and Vangelis Karkaletsis. *NewSum: "N-Gram Graph"-Based Summarization in the Real World*. IGI, 2014.

[Giannakopoulos *et al.*, 2015] George Giannakopoulos, Jeff Kubina, Ft Meade, John M Conroy, MD Bowie, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. Multiling 2015: Multilingual summarization of single and multi-documents, online fora, and call-center conversations. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 270, 2015.

[Kabadjov *et al.*, 2015] Mijail Kabadjov, Josef Steinberger, Emma Barker, Udo Kruschwitz, and Massimo Poesio. Onforums: The shared task on online forum summarisation at multiling15. *Analysis*, 16:23, 2015.

[Luhn, 1958] H. P Luhn. Automatic creation of literature abstracts, the. *IBM Journal of Research and Development*, 2(2):159165, 1958.

[Mihalcea, 2005] R. Mihalcea. Multi-document summarization with iterative graph-based algorithms. In *Proceedings of the First International Conference on Intelligent Analysis Methods and Tools (IA 2005)*. McLean, 2005.

[Qazvinian *et al.*, 2013] Vahed Qazvinian, Dragomir R. Radev, Saif M. Mohammad, Bonnie Dorr, David Zajic, Michael Whidby, and Taesun Moon. Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, 46:165201, 2013.

[Swanson *et al.*, 2015] Reid Swanson, Brian Ecker, and Marilyn Walker. Argument mining: Extracting arguments from online dialogue. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 217, 2015.