xLiMe
FP7-ICT-2013-10
ICT-2013.4.1
Contract no.: 611346
www.xlime.eu

Deliverable D5.2.1

Early Opinion Diffusion Prototype

Editor:	Blaž Novak, JSI
Author(s):	Blaž Novak, JSI; Aljaž Košmerlj, JSI; Kristjan Voje, JSI
Deliverable Nature:	Report (R)
Dissemination Level:	Public (PU)
Contractual Delivery Date:	M24 – 31 October 2015
Actual Delivery Date:	M24 – 31 October 2015
Suggested Readers:	All project partners
Version:	1.0
Keywords:	Information diffusion; networks; news, sentiment

Disclaimer

Full Project Title:	xLiMe – crossLingual crossMedia knowledge extraction
Short Project Title:	xLiMe
Number and Title of Work package:	WP5 Cross-lingual, Cross-media Analytics and Reporting
Document Title:	D5.2.1 – Early Opinion Diffusion Prototype
Editor:	Blaž Novak, JSI
Work package Leader:	Blaž Novak, JSI

Executive Summary

Task T5.2 deals with analysis and modelling of the diffusion of information between content creators, through various modalities, languages and geographic locales.

In this deliverable we describe the early prototype of the system that we developed, which currently processes news and TV content available to the project, and maintains a continuously updated model of how information spreads throughout the observed news publishing world. The system provides an API through which the information about actors in the knowledge diffusion process can be analysed and retrieved. We currently model information diffusion by monitoring evidence of text copying, and by tracking the sentiment of documents in diffusion cascades.

Cross-modal analysis is already supported by treating TV subtitles and automatically recognized speech the same as documents produced by TV programmes. Analysis of information in multiple languages is also already supported, but proper cross-lingual analysis isn't yet fully functional. While the current system can process social media input, it is not yet optimized for it.

As a side effect of the model, additional information about actors is made available to the rest of the project pipeline.

The system is integrated with the rest of the project pipeline on its input side, while the output is temporarily provided as a stand-alone HTTP API until exact data and format requirements are determined by use-cases.

Table of Contents

Abbreviations

API	Application Programming Interface
FSM	Frequent sequence mining
HTTP	Hyper-Text Transfer Protocol
IDF	Inverse Document Frequency
JSON	JavaScript Object Notation

1 Introduction

The large amount of redundancy in the published news content on the internet provides us with an opportunity for analysis of how the information is simultaneously spread through the internet and traditional media, such as broadcast TV.

Unlike other published research that deals with diffusion of information through networks, we have an almost unlimited amount of accessible and strongly duplicated content, with similar topics, entities and phrases occurring in multiple modalities and languages at the same time.

We want to be able to monitor how publishers – either web sites or TV programmes, but ultimately also individuals on social networks – influence each other. The first question we want to answer is which authors are likely to produce original content, which ones tend to follow by summarizing or even plagiarizing it, and who has access to factual information on a given topic.

We assume that publishers are complex entities composed of multiple more or less independent authors, and that a simple global model will not completely describe the information diffusion. However, we find that a rather simple model gives useful insights.

We also want to be able to model what happens to the sentiment about a certain event as the news of it spreads through the world, and predict sentiment (dis)agreement in the diffusion chain, with a potentially topic-dependant model.

We want to be able to abstract the information according to various parameters, for example to focus on how information and sentiment spread between languages, between geographic locales, between media modalities, etc.

We would also like to monitor the evolution of the diffusion model through time, to expose changing attitudes of authors through time.

So far, we have developed modules for extraction of evidence of information diffusion between publishers, multiple models for prediction of likelihood of information and sentiment diffusion, and some supporting functionality.

The model is updated in real-time from both TV and news data, and is accessible through an API, which allows the consumer to see how a certain author participates in the diffusion network, and what his sentiment towards certain topics is. API queries can specify time periods of interest, which for now allows the end-user to manually analyse the evolution of the model.

TV content that is provided by Zattoo is analysed by our system developed in WP2 and is described in deliverable D2.1.2 [1]. It produces a stream of RDF messages in the project Kafka message queue, that describe 40 second chunks of TV content. Two different types of messages are published to Kafka – one for subtitles, if they are provided by content creators, and the other for text automatically extracted from audio channel.

We use a pair of whitelists to select channels and programs which we consider to be news content. This allows us to ignore speech from channels which are being transcribed for other use cases. We created whitelists manually by picking recurring TV program names which were marked with “informational content” tag.

News content is provided by the JSI Newsfeed [2] system. Both TV and news text are fed into the JSI EventRegistry [3] system. This uses technology developed during the XLike project [4] to match news articles together into semantically coherent events. Related articles in multiple languages are all assigned to the same event.

In the xLiMe project, we have extended this system to be capable of ingesting text from other sources and process it in the same manner as internet news articles. TV content is seamlessly integrated into existing

events, which means that no special handling is needed to analyse it. Depending on the desired use, either a TV channel or a specific TV program can be considered to be the publisher of the content.

Information about detected events is exported as an RDF semantic graph to the Kafka pipeline, as described in the deliverable D4.3.1 [5]. This includes general information about detected events, documents that belong to specific events, their source, full text, language, detected concepts, and categories they belong to.

We process the event information by first extracting evidence for information diffusion. We annotate all the participating articles with a sentiment detection model and transfer the annotations to the diffusion evidence.

The entire set of found evidence is simplified according to the selected focus – for example according to publisher identities, geography, or source modality. This is used to build a graph that models the likelihood of information transfer, and sentiment correlation.

Alternatively, evidence is grouped together by events, and processed to extract discrete paths of information diffusion within each independent event. A frequent sequence mining algorithm is used to discard sequences with insufficient support. These sequences can be calculated by either including the sentiment information, or not.

At the same time, we also keep track of sentiment associated with named entities that are mentioned by specific articles.

Currently, we use entire articles as basic units of information diffusion tracking and sentiment detection; we plan on evaluating if going to paragraph level improves the model.

2 Diffusion evidence extraction

Figures 1 to 4 show the kind of information extracted in the first stage of the pipeline.

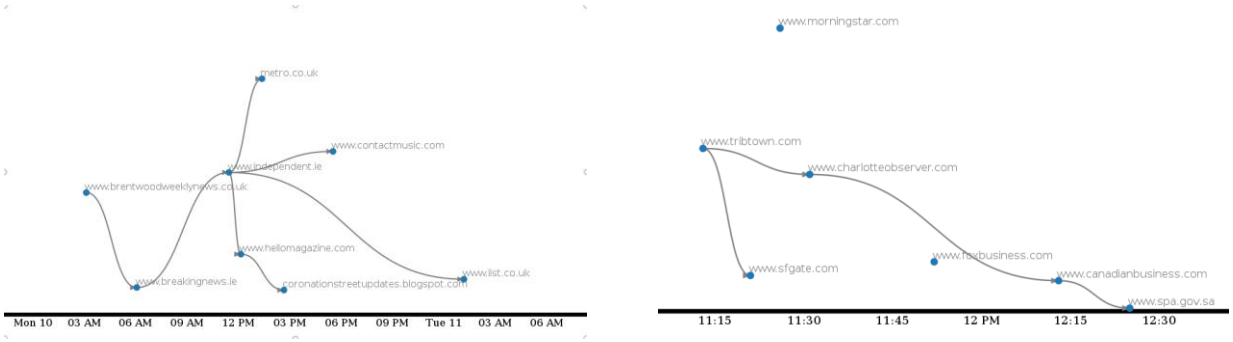

Figure 1, 2: Event information diffusion subgraph example 1 and 2

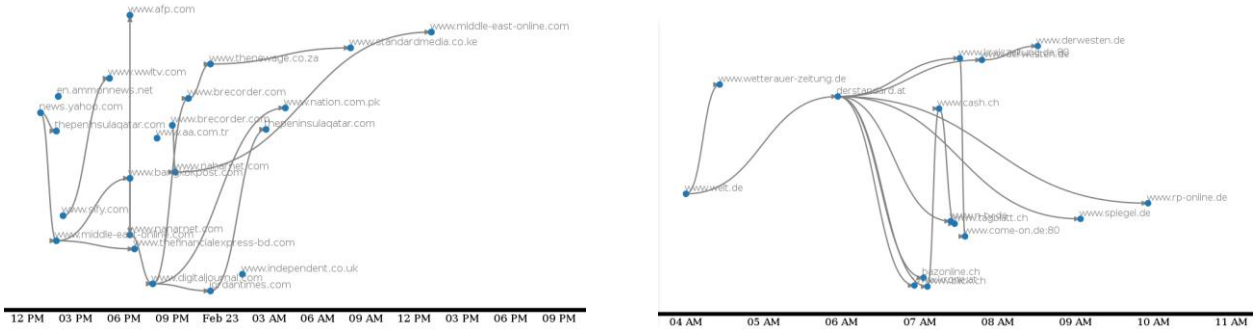

Figure 3, 4: Event information diffusion subgraph example 3 and 4

Our initial algorithm works on the level of individual events as produced by EventRegistry. EventRegistry matches documents together using a clustering algorithm that has access to all of the articles published in a given time window. It also takes into account cross-lingual similarity between articles. We use this partitioning of the set of articles to restrict extracted diffusion evidence to those cases where both source and destination come from the same event. This reduces the amount of computational work that needs to be done, and the amount of noise.

For all time ordered pairs of documents in an event, a similarity function is evaluated. We are currently using a normalized set overlap between word n-grams generated from both documents as the similarity, which is useful for detecting identical sequences of words.

Parameters of n-gram generation – maximum length, normalization and weighting – are configurable, but are set to some fixed value before the start of the pipeline. We have tried both simple set overlap and weighting n-grams by their inverse document frequency (IDF) calculated on either single events or a union of a large set of events – calculated by windowing the past event stream – and compared the calculated similarities on a sample of 10000 articles. We found that all variants produce similar orderings of article similarities, so we picked the one with IDF weighting based on multiple event n-gram statistics. All the similarity functions will be evaluated for the final prototype.

N-gram based similarity detects cases of direct phrase copying, which works surprisingly well, but is not suited for more subtle cases of information diffusion, or diffusion of information across languages. The JSI Newsfeed provides approximately 450.000 articles per day, of which about half are processed by EventRegistry. This redundancy in information might still make it possible to detect influence between publishers using a simple similarity function even when they avoid copying text directly.

Finding better similarity functions is currently work in progress.

Similarities for all pairs of articles are recorded along their event as similarity records. The images above show time ordered graphs of publishers that published an article in the corresponding event, connected by arrows in cases where the articles were determined to be connected according to the procedure described in the next section.

3 Evidence aggregation

When the first stage of the pipeline processes a new event and produces a set of article similarity records, they are used to update multiple information diffusion models.

3.1 Global network model

The first model we are using is a graph, where nodes represent publishers, languages, or geographic locations, depending on the type of analysis required. Similarity between articles *s* and *t*, where *s* was published before *t* is used as an indication that the publisher, language or country that produced article *s* influences the publisher, language, or country that produced article *t*. On a single article level, this is not very strong evidence, but patterns emerge when aggregated over the entire dataset.

Edges in the graph represent the influence of node *s* on node *t*. The influence is reduced with age of evidence. Two aging modes are implemented: averaging and exponential decay. Averaging is not really aging – weight of an edge is simply calculated as the average of all accumulated evidence, and as new evidence comes in, importance of old evidence decreases.

Exponential decay, on the other hand, ages the evidence for edge weight as an exponential function $e^{-\beta t}$, where β represents the aging rate. This can be easily implemented by keeping track of the last time an edge was updated, and multiplying its weight by $e^{-\beta(T_{now}-T_{last})}$ before adding the new evidence weight to it, without having to age the entire graph all the time.

We store periodic snapshots of a sparse version of the graph, to enable clients to calculate the difference between two points in time. Only a fixed number of edges of the graph are stored due to storage space constraints. In case an exact difference is required, a sufficiently long archive of events must replayed on a new graph. The size of the archival data required depends on the ageing constant. This operation is not yet exported by the API.

3.1.1 Sentiment diffusion network model

The described model can be extended to include basic information about sentiment diffusion, by augmenting graph edges with sentiment correlation information. The source and target node document sentiment values are calculated using an external sentiment classification model by the evidence generation stage. The current implementation produces a single evidence record for a pair of documents, so sentiment is calculated on source and target documents. We will evaluate the possibility of using paragraph level evidence generation, in which case we will calculate sentiment on the paragraph level, too.

As we are using a binary sentiment model, we maintain a matrix where one cell contains number of cases where both source and target documents had positive sentiment, one cell contains the number of cases where source document sentiment was negative, and target positive, etc. We will later use this linear model to predict the sentiment of an article in an information diffusion cascade based on sentiments of articles published by other publishers, and detect deviations from model predictions. This sentiment diffusion model is aged in the same way as the edge weight – as described in the previous section.

3.2 Frequent sequence mining

We have implemented an alternative diffusion model by counting discrete paths in sparse graphs induced by the calculated similarity records.

The algorithm reduces the list of similarity pairs attached to an event by discarding pairs with low similarity scores. This is done on a per-article level by removing all but a predefined number of similarity records, on

event level, by removing similarity records that fall below a certain threshold by absolute value and those that are lower than a user defined fraction of all records, and likewise on the entire dataset.

The remaining time ordered similarity records induce a (possibly disconnected) directed acyclic graph for each event in the EventRegistry database. By using a modified APRIORI[7] algorithm for frequent itemset mining, we extract a set of sequences of nodes (i.e. publishers, languages or geographical locales), such that each sequence, when interpreted as a path through a graph induced on the similarity record set, occurs more than a specified number of times in the entire dataset. When calculating the support for a sequence, all possible paths through all graphs of all events are counted.

Unlike the global network model, this one does not assume a single diffusion influence value between two nodes – there can be an arbitrary number of paths through a node. Each sequence is associated with a set of supporting events, which can be used by the client to create a topic model for that sequence.

Because the calculation of frequent sequences needs to be re-run for a selected time interval and is not a real-time operation, we currently do not support specifying time intervals for this model in the API.

3.2.1 Sentiment diffusion frequent sequence model

To include the sentiment information in the frequent sequence model, we append the sentiment of an article to the name of the node in the graph. Instead of a node “N”, the frequent sequence mining algorithm distinguishes between nodes “N+” and “N-“, which stands for “N, with positive sentiment”, and “N, with negative sentiment”.

Frequent paths extracted from the dataset using the same algorithm as in the preceding section thus already contain sentiment information.

3.3 Named entity sentiment model

Since the system has access to the list of detected named entities in all the documents, and the calculated sentiment value for each document, we have created a supplementary service that maintains for each named entity and for each publisher the average sentiment of all documents published by this publisher that mention the specified named entity.

The API client can retrieve these lists for a selected time interval to compare publishers by their attitude towards certain concepts, and to track its evolution through time.

4 Access to model state

The information that is maintained by the system can be queried using an HTTP API, with the endpoint address of http://xlime.ijs.si/t52/api. Currently, only the publisher-level granularity graph is accessible, with geography and language graphs to come in near future.

All of the query parameters which are listed in Table 1 are passed as an URL encoded string. The response is returned as a JSON object that contains the query parameters and the response.

Three main objects in the response are

1. pair of lists of nodes (i.e. publishers/..) in the directed graph that precede and follow the selected node, with the associated weights and sentiment transition likelihoods,
2. the annotated list of all frequent sequences of which the selected node is a member of, and
3. a list of named entities for which the current publisher holds a significantly non-average sentiment.

The result format details are not yet stable and should be verified before issuing API calls.

Parameter	Description
subject	A string representation of the object of interest. For publishers, it should be prefixed with “pub:”, e.g. “pub:www.reuters.com”, for languages with “lang:” and for countries with “geo:”.
start_time	Beginning of the interval of interest, for which the query is to be executed. Does not influence the output from frequent sequence mining module. Format: YYYY-MM-DDTHH-MM-SSZ
end_time	End of the interval of interest in the same format as start_time.

Table 1: Description of API parameters

5 Related work

There has been a lot of work done on the topic of information diffusion on networks in the recent years. Most of research has been done on social networks, mainly Twitter and Facebook, with a smaller share of blog analysis. Papers by [7] and [8] give a survey of research on information diffusion, opinion mining and sentiment analysis. We provide a short description of some other related research here. While it does not directly deal with multi-modal information in all but one case, it covers various aspects of opinion mining.

Li et al. [9, 10] introduce a theoretical framework of opinion diffusion in social networks, where they model opinions of an agent as a vector of probabilities that individual facts are true. Their analysis is theoretical - they don't use real-world data, only simulations.

Parsegov et al. present a similar framework as Li et al. in [11], but also model interactions between individual belief topics. They also only use simulations to verify their framework.

A theoretical analysis of opinion dynamics in online social networks through modelling three types of interaction – agreement, antagonism, and neutrality between nodes in a directed graph is given by Wang et al in [12].

Bakshy et al. performed a large-scale (253 million Facebook users) field experiment in which they analysed the role of strong and weak links in information propagation in a social network, and described it in [13].

Theory of infectious diseases was applied to the problem of information diffusion by Gruhl et al. in [14]. Yang and Leskovec [15] developed a model of global information diffusion influence of a node without explicitly modelling the underlying social network.

Unsupervised discovery of opposing opinion networks in online forums was done by Lu et al. in [16]. They used both textual and network features, and present results on a dataset of manually annotated forum posts, on a couple of controversial topics.

Quattrociocchi et al. have created a theoretical model of opinion dynamics that accounts for the coexistence of other media modalities (such as TV and newspapers) and of social influence as two separated but interdependent processes in [17], however they also only perform simulated experiments.

A paper by Wu et al. [18] describes OpinionFlow, a visual analysis system for detection and analysis of opinion propagation patterns in online social networks, and presents results obtained by analysing opinion diffusion between Twitter users.

6 Conclusion and future work

In the task T5.2, we have implemented a prototype system that tracks news articles and TV broadcast news content, and analyses it for patterns of influence between publishers. Combined with the semantic graph of events provided by task T4.3, it will be used to provide insights into opinion diffusion in the monitored network when it is extended in the upcoming year.

The system is already receiving data from the project pipeline, but is not yet fully integrated due to the lack of requirement specifications, which we will also produce. Data that is generated by the system is available to consumers through an HTTP API.

We have already identified multiple directions for future work in previous chapters. Most importantly, a good evaluation environment needs to be set up in order for us to be able to test different approaches. Evaluation metrics will likely depend on the use case requirements.

References

[1] D2.1.2 – Final speech to text prototype. xLiMe deliverable. Blaz Novak, JSI, 2015.

[2] http://newsfeed.ijs.si/

[3] http://eventregistry.org/

[4] http://www.xlike.org/

[5] D4.3.1 – Early semantic graph construction. xLiMe deliverable. Aljaz Kosmerlj, JSI, 2015.

[6] Agrawal, Rakesh, Srikant, Ramakrishnan, "Fast algorithms for mining association rules in large databases". In Proceedings of the 20th International Conference on Very Large Databases, VLDB, 487-499, Santiago, Chile, Sep. 1994.

[7] Guille, Adrien, Hakim Hacid, Cécile Favre, and Djamel A. Zighed. "Information diffusion in online social networks: A survey." ACM SIGMOD Record 42, no. 2 (2013): 17-28.

[8] Liu, Bing, and Lei Zhang. "A survey of opinion mining and sentiment analysis." In Mining text data, pp. 415-463. Springer US, 2012.

[9] Li, Lin, Anna Scaglione, Ananthram Swami, and Qing Zhao. "Phase transition in opinion diffusion in social networks." In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 3073-3076. IEEE, 2012.

[10] Li, Lin, Anna Scaglione, Ananthram Swami, and Qing Zhao. "Trust, opinion diffusion and radicalization in social networks." In Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on, pp. 691-695. IEEE, 2011.

[11] Parsegov, Sergey E., Anton V. Proskurnikov, Roberto Tempo, and Noah E. Friedkin. "A Novel Multidimensional Model of Opinion Dynamics in Social Networks." arXiv preprint arXiv:1505.04920 (2015).

[12] Wang, Shixiong, and Yi Jiang. "Extending opinion dynamics model for collective online behaviours analysis." In Computer Modelling & New Technologies. Volume 18(11), pp 914-920, 2014.

[13] Bakshy, Eytan, Itamar Rosenn, Cameron Marlow, and Lada Adamic. "The role of social networks in information diffusion." In Proceedings of the 21st international conference on World Wide Web, pp. 519-528. ACM, 2012.

[14] Gruhl, Daniel, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. "Information diffusion through blogspace." In Proceedings of the 13th international conference on World Wide Web, pp. 491-501. ACM, 2004.

[15] Yang, Jaewon, and Jure Leskovec. "Modeling information diffusion in implicit networks." In Data Mining (ICDM), 2010 IEEE 10th International Conference on, pp. 599-608. IEEE, 2010.

[16] Lu, Yue, Hongning Wang, ChengXiang Zhai, and Dan Roth. "Unsupervised discovery of opposing opinion networks from forum discussions." In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 1642-1646. ACM, 2012.

[17] Quattrociocchi, Walter, Guido Caldarelli, and Antonio Scala. "Opinion dynamics on interacting networks: media competition and social influence." Scientific reports 4 (2014).

[18] Wu, Yingcai, Shixia Liu, Kai Yan, Mengchen Liu, and Fangzhao Wu. "OpinionFlow: Visual analysis of opinion diffusion on social media." Visualization and Computer Graphics, IEEE Transactions on 20, no. 12 (2014): 1763-1772.

