# Generating Semantic Snapshots of Newscasts using Entity Expansion

José Luis Redondo García[1], Giuseppe Rizzo[1], Lilia Perez Romero[2], Michiel Hildebrand[2], Raphaël Troncy[1]

[1] EURECOM, Sophia Antipolis, France,
{redondo, giuseppe.rizzo, raphael.troncy}@eurecom.fr
[2] CWI, Amsterdam, The Netherlands,
{L.Perez, M.Hildebrand}@cwi.nl

**Abstract.** TV newscasts report about the latest event-related facts occurring in the world. But relying exclusively on them is therefore insufficient to fully grasp the context of the fact being reported. In this paper we propose an approach that retrieves and analyzes related documents in the Web to automatically generate semantic annotations that provide viewers and experts comprehensive information about the news. Named entities detected in the retrieved documents further disclose relevant concepts that were not explicitly mentioned in the original newscast. A ranking algorithm based on entity frequency, popularity peak analysis, and domain experts rules sorts those annotations to generate the Semantic Snapshot of the considered Newscast (NSS). We benchmark this method against a gold standard generated by domain experts and assessed via a user survey over five BBC newscasts. Results of the experiments show the robustness of the approach holding an Average Normalized Discounted Cumulative Gain of 66.6%.

**Keywords:** Semantic Video Annotation, Entity Expansion, Newscasts

## 1 Introduction

With the emergence of both citizen-based and social media, traditional information TV channels have to re-think their production and distribution workflow processes. We live in a globalized world, a vast playing field where events happening are the result of complex interactions between many diverse agents along time. The interpretation of those news is problematic because of two issues. *i)* the *need of background*: viewers often need to be aware of other facts that happened in a different temporal or geographic dimension. *ii)* the *need of completeness*: a single representation of an event is not enough to capture the whole picture, because it is normally incomplete, it can be biased, or partially wrong. Some TV applications assist viewers in consuming those news, but they still need to be fed with meaningful details concerning a news item. The most common strategy to get this information is to enrich the original content with additional data collected from external sources. However, this results in large amounts of unreliable

and repeated information, leaving to the user the burden of processing the large amounts of potentially related data to build an understanding of the event.

One strategy reported in the literature for having such a mechanism is to perform named entity extraction over the newscast transcript [6]. However, the set of named entities obtained from such an operation is insufficient and incomplete for expressing the context of a news event [4]. Sometimes entities spotted over a document are not disambiguated because the textual clues surrounding the entity are not precise enough. While in some cases, some relevant entities are not mentioned in the transcripts. In this paper we automatically retrieve and analyze additional documents from the Web where the same event is also described, in a process called Newscast Named Entity Expansion. By increasing the size of the document set to analyze, we increase the completeness of the context and the representativeness of the list of entities, reinforcing relevant entities and finding new ones that are potentially interesting. This approach is able to produce a ranked list of entities called Newscast Semantic Snapshot (NSS), which includes the initial set of detected entities in subtitles with other event-related entities captured from the seed documents.

The paper is organized as follows: Section 2 presents the related works, Section 3 illustrates the approach depicted in this paper for generating NSS. Section 4 describes in depth the different ranking algorithms used for ordering the list of candidate entities. The experimental settings are described in Section 5. We summarize the main findings and future plans in Section 6.

## 2 Related Work

The need of a NSS for feeding certain applications is a concept we have already investigated in some previous research works and prototypes [7], proving the benefit of browsing the "surrounding context" of a newscasts. The same concept[3] was presented in the Iberoamerican Biennial of Design (BID).[4] with great feedback from users and experts. In the domain of Social Networks, named entities are used for identifying and modeling events and detecting breaking news. In [10], the authors emphasize the importance of spotting entities in short user generated posts in order to better understand their topic. Entities have been used for video classification when the textual information attached to a video contains temporal references (e.g. subtitles) [5].

In the literature there are some approaches relying in similar expansion techniques. Set expansion using the Web has been applied to the problem of unsupervised relation learning [1], deriving features for concept-learning [2], or computing similarity between attribute values in autonomous databases [15]. In [12], authors proposed a system called SEAL (Set Expander for Any Language). SEAL works by automatically finding semi-structured web pages that contain lists of items and aggregating these lists in a way that the most promising items are ranked higher. The same authors published an improved version

[3] https://vimeo.com/119107849

[4] http://www.bid-dimad.org

of the algorithm [13], increasing the performance by expanding a couple of randomly selected seeds and accumulating information along different iterations. Our approach focuses on maximizing the quality of a single search query for obtaining the most appropriate set of related documents to be analyzed. Another approach that works extending a set of entities is [11], which combines the power of semantic relations between language terms like synonymy and hyponymy and grammar rules in order to find additional entities in the Web sharing the same category that the ones provides as input. To the best of our knowledge, the only related work in the news domain that has been carried out grounding the power of enriching the set of initial entities by using an entity expansion algorithm is a previous paper of ours [8]. It includes a naive document collection strategy with no filtering of entities, it proposes an entity ranking algorithm based on the appearance of the entities in the collected documents, and exploits the DBpedia knowledge base as a way to ensure the coherence of the final list of entities. The work presented in this paper improves the referred work in several directions: document retrieval mechanism, semantic annotation, creation of the NSS.

## 3 Newscast Entity Expansion Approach

Our approach for generating Newscast Semantic Snapshots is composed of five main steps: query formulation, document retrieval, semantic annotation, annotation filtering, and annotation ranking. Fig. 1 depicts the expansion process.

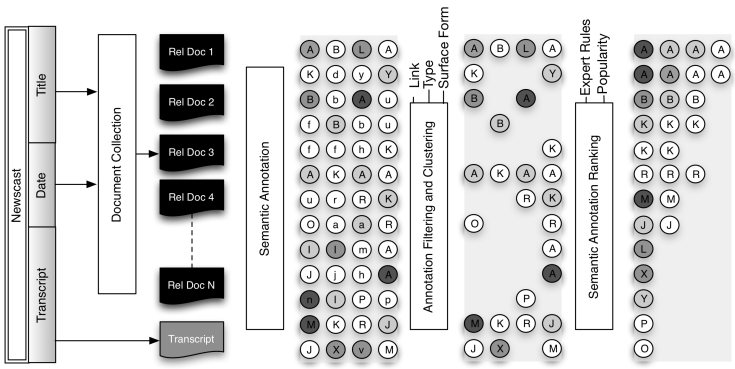


**Fig. 1:** Schema of Named Entity Expansion Algorithm.

**Query Formulation** Newscast broadcasters offer metadata about the items they publish, which is normally available together with the audiovisual content itself. In this work, we build the query $q = [\mathrm{h}, \mathrm{t}]$, where $h$ is the video heading, and $t$ is the publication date.

**Document Retrieval** The retrieval stage has the intent to collect event-related documents from the open Web as result of the query $q$. It selects a set of the documents $D$ over which the semantic annotation process is performed.

The quality and adequacy of the collected documents sets a theoretical limit on how good the resulting news annotations are.

**Semantic Annotation** In this stage we perform a named entity recognition analysis with the objective of reducing the cardinality of the textual content from the set $D$ of documents $\{d_1, ..., d_n, d_{n+1}\}$ where $d_{n+1}$ refers to the original newscast transcript. HTML tags and other annotations are removed. The feature space is then reduced and each document $d_i$ is represented by a bag of entities $E_{d_i} = e_{1_{d_i}}, ..., e_{n_{d_i}}$, where each entity is defined as a triplet $(surface\_form, type, link)$.

**Annotation Filtering and Clustering** The Document Retrieval stage expands the content niche of the newscast. At this stage we apply coarse-grained filtering of the annotations $E$ obtained from the previous stage, applying a $f(E_{d_i}) \rightarrow E'_{d_i}$ where $\left|E'_{d_i}\right| < |E_{d_i}|$. The particular strategies used will be further explained with the experimental settings in Section 5. Named entities are then clustered applying a centroid-based clustering operation based on strict string similarity over the *link*, and in case of mismatch, the Jaro-Winkler string distance [14] over the $surface\_form$. The output of this phase is a list of clusters containing different instances of the same entity.

**Semantic Annotation Ranking** The bag of named entities $E'_{d_i}$ is further processed to promote the named entities which are highly related to the underlined event. We propose a ranking strategy based on entity appearance in documents, popularity peak analysis, and domain experts' rules sorts the annotations to generate the Semantic Snapshot of the considered Newscast (NSS).

## 4 Ranking Strategy

The unordered list of entities is ranked to promote those that are potentially interesting for the viewer. The strategies presented below ground on the assumption that entities appearing often in the retrieved documents are more important. We propose two different functions for scoring the frequency of the entities. We then consider two orthogonal functions which exploit the entity popularity in the event time window and the domain experts' knowledge.

### 4.1 Frequency-based Function

We first rank the entities according to their absolute frequency within the set of retrieved documents $D$. Defining the absolute frequency of the entity $e_i$ in a collection of documents $D$ as $f_a(e_i, D)$, we consider the scoring function $S_F = \frac{f_a(e_i,D)}{|E|}$, where $|E|$ is the cardinality of all entities across all documents. In Fig. 2 (a) we can observe how entities with high $S_F$ are on the right side of the plot and become part of the NSS.

### 4.2 Gaussian-based Function

The $S_F$ function privileges entities which appear the most. However from a viewer's perspective these frequent entities often represent concepts that have

been so present in media that have become evident to them. To approximate this scoring strategy, we relied on a Gaussian curve which penalized both lowly and highly repeated entities. By characterizing the entities in terms of their Bernoulli appearance rate across all documents $f_{doc}(e_i)$ and applying the Gaussian distribution over those values, we promote entities distributed around the mean $\mu = \frac{|D|}{2}$ via the function $S_G = 1 - \left| \frac{f_{doc}(e_i)}{|D|} - 1 \right|$ (Fig. 2 (b)).

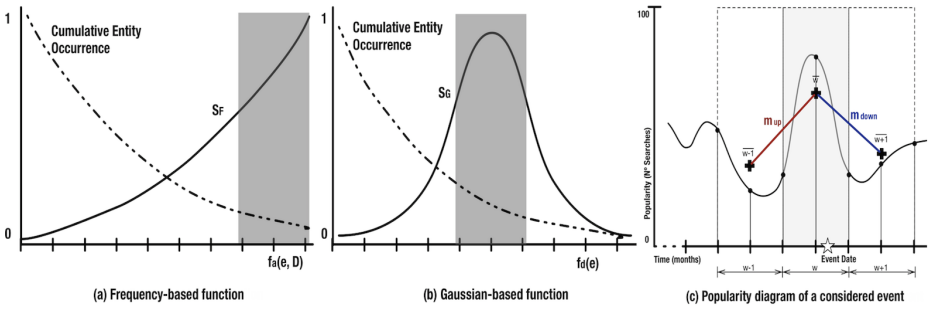


**Fig. 2:** (a) depicts the Decay function of the entity occurrences in the corpus, and the $S_F$, (b) represents the Gaussian-based function $S_G$, with the entities highly important over the mean, and (c) the calculation of the popularity score for a considered event

### 4.3 Orthogonal Functions

**Popularity Function** We propose a function that considers variations in entity popularity values (commonly named as popularity peaks) around the date of the studied event. Pure frequency based approaches fail to promote certain entities which are barely mentioned in related documents but eventualy become interesting for viewers. The only solution is to rely on external sources providing indications about the entity popularity, like Google Trends[5] or Twitter[6].

The procedure for getting $P_{peak}(e_i)$ is depicted in Fig. 2 (c). Using the label of an entity $e_i$, we obtain a list of pairs $[t, P]$ where $P$ is the popularity score of an entity at the instant of time $t$. Afterward we create three consecutive and equally long temporal windows around $t$: $w_t$ containing the date itself, another one just immediately behind $w_{t-1}$ and a last one after the previous two $w_{t+1}$. In a next step we approximate the area inside the regions by calculating the average of the points contained in them, obtaining $\overline{w-1}$, $\overline{w}$ and $\overline{w+1}$. The variation in area between popularity under $\overline{w-1}$ and $\overline{w}$, and $\overline{w}$ and $\overline{w+1}$ give the values $m_{up}$ and $m_{down}$ respectively, which are normalized and combined into a single score for measuring how significant the increase in volume of searches was for that studied entity label. By empirically studying the distribution of the popularity scores of the entities belonging to a newscast, we have observed that

[5] https://www.google.com/trends

[6] https://twitter.com

it follows a Gaussian curve. This fact will help us to better filter out popularity scores that are not able to trigger valid conclusions.

**Expert Rules Function** The knowledge of experts in the domain like journalists can be materialized in the form of rules that correct the scoring output produced by former ranking strategies. The preconditions for activating those rules involve entity features such as type, number of documents where they appear, or the nature of the sources where the documents are coming from. Actions triggered consist of adjustments in the relevance score according to a factor $Op_{expert}$, which intends to recreate expert's judgements for promoting or penalizing those candidate entities: $S_{expert}(e) = S_{F-1}(e) * Op_{expert}$.

## 5 Experimental Settings and Evaluation

In this section we describe the experimental settings and we present the results of our evaluation against our golden standard. To the best of our knowledge there was no evaluation dataset suited to this context, so we have built our own. With this objective in mind, we selected 5 news videos and manually extracted entities from the subtitles; video image; text contained in the video; articles related to the subject of the video; and entities suggested by an expert. After building a candidate set of entities, we presented this set to 50 participants via an online survey and asked them to rate their level of interestingness. The methodology for building this dataset is available online at `https://github.com/jluisred/NewsEntities`, together with the list of entities and scores per video.

Inspired by studies in Web search engines, we have based our evaluation procedure in a measure called Average Normalized Discounted Cumulative Gain ($MNDCG$ at $N$), which is based on the assumption that resulting documents can have different degrees of relevance for one particular task [3]. As the relevant documents in our gold standard are scored according to users' relevance, this measure can provide a more exhaustive judgment about the adequacy of the generated NSS. Concerning N, we have empirically studied the whole set of queries and main ranking functions observing that values of $MNDCG$ decreasingly improve until they reach a stable behavior from $N = 10$ on.

### 5.1 Experimental Settings

**Document retrieval** We have relied on the Google Custom Search Engine (CSE) API service[7] by launching a query$q = [\mathrm{h}, \mathrm{t}]$. The maximum number of retrieved document is set to 50. The CSE engine also considers other parameters that need to be tuned up:

1. Web sites to be crawled. We have considered five possible values: search over the whole set of Web pages indexed by Google, search over a set of 10 internationals English speaking newspapers[8] (*L1*), search in the set of 3

[7] `https://www.google.com/cse/all`

[8] `http://en.wikipedia.org/wiki/List_of_newspapers_in_the_world_by_circulation`

international newspapers used in the gold standard creation (*L2*), prioritize results from in *L1* but still consider other sites, and prioritize content in *L2* but still consider other sites.

2. Temporal dimension. We consider the time window $T_{Window} = [\mathrm{t} - d, \mathrm{t} + d]$, with two possible values for $d$: 3 days and one week.
3. In addition, Google CSE can filter results according to the Schema.org types. In our experiments we tried with [NoFilter, Person&OrganizationFiltering]

**Semantic Annotation** We use [9] which applies machine learning classification of the entity type given a rich feature vector composed of a set of linguistic features, the output of a properly trained Conditional Random Fields classifier and the output of a set of off-the-shelf NER extractors supported by the NERD Framework.[9] We used it as an off-the-shelf entity extractor, using the offered classification model trained over the newswire content.

**Annotation Filtering and Clustering** In order to get rid of some non-pure named entities which are not well considered by viewers and experts, we have applied three different filtering approaches: filtering according to their NERD type[10] (*F1*), in our case, we keep only Person, Location, and Organization, removing entities with confidence score under first quarter of the distribution (*F2*), and keeping only entities with capitalized surface form (*F3*). A first preselection by setting to default the rest of steps of the approach led us to discover that 3 of the filters (*F1*, *F3*, and combination *F1+F3*) were producing best MNDCG values.

**Semantic Annotation Ranking** For the current experiment we run both Frequency and Gaussian based functions, together with the orthogonal strategies based on popularity and expert rules. Regarding the *popularity* dimension, we have relied on Google Trends, which estimates how many times a search-term has been used in a given time-window. Since Google Trends gives results with a monthly temporal granularity, we have fixed the duration of $w$ to 2 months to keep the sample representative. With the aim of keeping only those findings backed by strong evidence, we have filtered the entities with peak popularity value higher than $\mu + 2 * \sigma$. Those entities will adjust their former scores with the popularity values via the following equation: $S_P(e) = R_{score}(e) + Pop_{peak}(e)^2$. Concerning the *Expert Rules* dimension, we have considered three rules to be applied over the three main entity types. The different $Op_{expert}$ values have been deduced by relying on the average score per entity type computed in the survey. Organizations have gotten a higher weight (0.95), followed by Persons (0.74), and by Locations (0.48) that are badly considered and therefore lower ranked in general. A last rule applied over entities appearing in less than two sources $f_{doc}(e_i) < 2$ automatically discards the matched instances ($Op_{expert} = 0$).

[9] `http://nerd.eurecom.fr`

[10] `http://nerd.eurecom.fr/ontology`

### 5.2 Results

Given the different settings for each phase of the approach we have a total of $20 * 4 * 4 = 320$ different runs that have ranked according to $MNDCG_{10}$. In addition we have considered two baselines:

**Baseline 1 (BS1):** Former Entity Expansion Implementation. A previous version of the News Entity Expansion algorithm was already published in [8]. The settings are: Google as source of documents, temporal window of 2 Weeks, no Schema.org selected, no filter strategy applied, and only frequency based ranked function with no orthogonal appliances.

**Baseline 2 (BS2):** TFIDF-based Function. To compare our functions with other traditional frequency based approaches, we selected the well-known TF-IDF. It measures the importance an entity in a document over a corpus of documents $D$, penalizing those entities appearing more frequently along the whole set of documents.

$$tf(e_i, d_j) = 0.5 + \frac{0.5 \times f_a(e_i, D)}{max\{f_a(e'_i, D) : e'_i \in d_j\}}, idf(e_i, d_j) = log \frac{|D|}{\{d_j \in D : e_i \in d_j\}} \quad (1)$$

We aggregated the different $tf(e_i, d_j) \times idf(e_i, d_j)$ into a single score via the function $S_{TFIDF}(e) = \frac{\sum_{j=1}^{n} tf(e,d_j) \times idf(e)}{|D|}$.

**Table 1:** Executed runs and their configuration settings, ranked by $MNDCG_{10}$

| Run | Collection | | | Filtering | Functions | | | Result | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sources | $T_{Window}$ | Schema.org | | Freq | Pop | Exp | $MNDCG_{10}$ | $MAP_{10}$ | $MP_{10}$ | $MR_{10}$ |
| Ex0 | Google | 2W | | F1+F3 | Freq | | ✓ | 0.666 | 0.71 | 0.7 | 0.37 |
| Ex1 | Google | 2W | | F3 | Freq | | ✓ | 0.661 | 0.72 | 0.68 | 0.36 |
| Ex2 | Google | 2W | | F3 | Freq | ✓ | ✓ | 0.658 | 0.64 | 0.6 | 0.32 |
| Ex3 | Google | 2W | | F3 | Freq | | | 0.641 | 0.72 | 0.74 | 0.39 |
| Ex4 | L1+Google | 2W | | F3 | Freq | | ✓ | 0.636 | 0.71 | 0.72 | 0.37 |
| Ex5 | L2+Google | 2W | | F3 | Freq | | ✓ | 0.636 | 0.72 | 0.7 | 0.36 |
| Ex6 | Google | 2W | | F1+F3 | Freq | | | 0.626 | 0.73 | 0.7 | 0.38 |
| Ex7 | L2+Google | 2W | | F3 | Freq | | | 0.626 | 0.72 | 0.72 | 0.37 |
| Ex8 | Google | 2W | | F1+F3 | Freq | ✓ | ✓ | 0.626 | 0.64 | 0.56 | 0.28 |
| Ex9 | L2+Google | 2W | | F1+F3 | Freq | | ✓ | 0.624 | 0.71 | 0.7 | 0.37 |
| Ex10 | Google | 2W | | F1 | Freq | | ✓ | 0.624 | 0.69 | 0.62 | 0.32 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| Ex78 | Google | 2W | ✓ | F1+F3 | Gaussian | | ✓ | 0.552 | 0.66 | 0.66 | 0.34 |
| Ex80 | L2+Google | 2W | ✓ | F1+F3 | Gaussian | | ✓ | 0.55 | 0.69 | 0.7 | 0.36 |
| Ex82 | L1 | 2W | ✓ | F3 | Gaussian | | ✓ | 0.549 | 0.68 | 0.64 | 0.33 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| BS2 | Google | 2W | | | Freq | | | 0.473 | 0.53 | 0.42 | 0.22 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| BS1 | Google | 2W | | | TFIDF | | | 0.063 | 0.08 | 0.06 | 0.03 |

In Table 1 we present the top 10 runs together with some lower configurations after position 78 and scores of the baseline strategies:

- Our best approach has obtained a $MNDCG_{10}$ score of 0.662 and a $MAP_{10}$ of 0.71, which are reasonably good in the document retrieval domain.
- Our approach performs much better than *BS1* and by far better than *BS2*. The very low score of this last baseline reveals that traditional TF-IDF

function is designed to measure the relevance of an item referred to the document that contains it and not the whole collection.
- Regarding the Document Retrieval step, using Google as source alone of together with other whitelists gives better results than restricting only to particular whitelists. The biggest $T_{Window}$ of 2 weeks performs better in all cases, while the use of Schema.org does not bring significant improvement except when applied over the Gaussian function (see runs 78, 80, 82).
- The best Filter strategy is *F3*, followed by the combination *F1_F3*. In conclusion, capitalization is a very powerful clue for refining the candidate list.
- Absolute frequency function performs better than Gaussian in all top cases.
- Expert Rules improves NSS for almost every configuration possible.
- Popularity based function does not seem to improve significantly the results. However, a further manual study of the promoted entities has revealed that the method is bringing up relevant entities like for example *David Ellsberg* for the query "Fugitive Edward Snowden applies for asylum in Russia". This entity is barely mentioned in the collected documents, but its role in the whole story is quite relevant[11], since he published an editorial with high media impact in The Guardian praising the actions of Snowden in revealing top-secret surveillance programs of the NSA.

## 6 Conclusion

In this paper we have presented an approach for automatically generating Newscast Semantic Snapshots. By following a entity expansion process that retrieves additional event-related documents from the Web, we have been able to enlarge the niche of initial newscast content. The bag of retrieved documents, together with the newscast transcript, is analyzed with the objective of extracting named entities referring to persons, organizations, and locations. By increasing the size of the document set, we have increased the completeness of the context and the representativeness of the list of entities, reinforcing relevant entities and finding new ones that are potentially interesting inside the context of that news item. We assessed the entire workflow against a gold standard, which is also proposed in this paper. The evaluation has showed the strength of this approach, holding an $MNDCG_{10}$ score of 0.666, outperforming the two studied baselines.

Future research interests include tailoring the entity ranking functions to particular news categories: sport, politics, business, etc. via supervised techniques (Learning to Rank). We are investigating how relations between entities specified in knowledge bases or inside the collected documents can be used to better rank them and even generate a graph based NSS. We also plan to use our approach as a means to suggest relevant entities in the process of the ground truth creation, in order to bring possible missing entities that were not identified because of human limitations in exhaustively covering the whole context of the news item.

[11] `http://en.wikipedia.org/wiki/Daniel_Ellsberg`

**Acknowledgments.** This work was partially supported by the European Union's 7th Framework Programme via the project LinkedTV (GA 287911).

## References

1. M. J. Cafarella, D. Downey, S. Soderland, and O. Etzioni. Knowitnow: Fast, scalable information extraction from the web. In *Human Language Technology Conference(HLT-EMNLP-05)*, pages 563–570, 2005.
2. W. W. Cohen. Automatically extracting features for concept learning from the web. In *17th International Conference on Machine Learning*, ICML '00, pages 159–166, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
3. W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.
4. M. Henzinger, B.-W. Chang, B. Milch, and S. Brin. Query-free news search. In *12th International Conference on World Wide Web*, WWW '03, pages 1–10, New York, NY, USA, 2003. ACM.
5. Y. Li, G. Rizzo, J. L. Redondo Garcia, and R. Troncy. Enriching media fragments with named entities for video classification. In *$1^{st}$ Worldwide Web Workshop on Linked Media (LiME'13)*, Rio de Janeiro, Brazil, 2013.
6. Y. Li, G. Rizzo, R. Troncy, M. Wald, and G. Wills. Creating enriched youtube media fragments with nerd using timed-text. In *11th International Semantic Web Conference (ISWC2012)*, November 2012.
7. J. Redondo-Garcia, M. Hildebrand, L. P. Romero, and R. Troncy. Augmenting tv newscasts via entity expansion. In *The Semantic Web: ESWC 2014 Satellite Events*, Lecture Notes in Computer Science, pages 472–476. Springer, 2014.
8. J. L. Redondo Garcia, L. De Vocht, R. Troncy, E. Mannens, and R. Van de Walle. Describing and contextualizing events in tv news show. In *23rd International Conference on World Wide Web Companion*, pages 759–764, 2014.
9. G. Rizzo, M. van Erp, and R. Troncy. Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web. In *$9^{th}$ International Conference on Language Resources and Evaluation (LREC'14)*, 2014.
10. T. Steiner, R. Verborgh, J. Gabarro Vallés, and R. Van de Walle. Adding meaning to social network microposts via multiple named entity disambiguation apis and tracking their data provenance. *International Journal of Computing Information Systems and Industrial Management*, 5:69–78, 2013.
11. M.-V. Tran, T.-T. Nguyen, T.-S. Nguyen, and H.-Q. Le. Automatic named entity set expansion using semantic rules and wrappers for unary relations. In *Asian Language Processing (IALP)*, pages 170–173, Dec 2010.
12. R. C. Wang and W. W. Cohen. Language-independent set expansion of named entities using the web. In *7th IEEE International Conference on Data Mining*, ICDM '07, pages 342–350, Washington, DC, USA, 2007.
13. R. C. Wang and W. W. Cohen. Iterative set expansion of named entities using the web. In *8th IEEE International Conference on Data Mining*, ICDM '08, pages 1091–1096, Washington, DC, USA, 2008.
14. W. E. Winkler. Overview of record linkage and current research directions. In *Bureau of the Census*, 2006.
15. G. Wolf, H. Khatri, B. Chokshi, J. Fan, Y. Chen, and S. Kambhampati. Query processing over incomplete autonomous databases. In *33rd International Conference on Very Large Data Bases*, VLDB '07, pages 651–662, 2007.