Using the past to explain the present: interlinking current affairs with archives via the Semantic Web

Yves Raimond, Michael Smethurst, Andrew McParland, Chris Lowis

BBC R&D, London, United Kingdom
firstname.lastname@bbc.co.uk

Abstract. The BBC has a very large archive of programmes, covering a wide range of topics. This archive holds a significant part of the BBC's institutional memory and is an important part of the cultural history of the United Kingdom and the rest of the world. These programmes, or parts of them, can help provide valuable context and background for current news events. However the BBC's archive catalogue is not a complete record of everything that was ever broadcast. For example, it excludes the BBC World Service, which has been broadcasting since 1932. This makes the discovery of content within these parts of the archive very difficult. In this paper we describe a system based on Semantic Web technologies which helps us to quickly locate content related to current news events within those parts of the BBC's archive with little or no pre-existing metadata. This system is driven by automated interlinking of archive content with the Semantic Web, user validations of the resulting data and topic extraction from live BBC News subtitles. The resulting interlinks between live news subtitles and the BBC's archive are used in a dynamic visualisation enabling users to quickly locate relevant content. This content can then be used by journalists and editors to provide historical context, background information and supporting content around current affairs.

1 Introduction

Large content archives can provide useful historical insights for current news events. For example a 2003 'Talking Point' episode on the BBC World Service dealing with the re-activation of a nuclear power plant in North Korea could provide some interesting background for a news story about North Korea's nuclear activity. A 1983 'Medical Programme' episode on techniques for measles immunisation or a 2000 'Science in Action' episode on predicting measles outbreaks can help to provide context around a recent epidemic.

The BBC (British Broadcasting Corporation) has broadcast radio programmes since 1922 and has accumulated a very large archive of programmes over the years. A significant part of this archive has been manually catalogued by professional archivists but the coverage of such metadata is not uniform across the BBC's archive. For example, it excludes the BBC World Service, which has been broadcasting since 1932. Little reuse is made of such parts of the BBC archives as there is little or no metadata to help locate content within them. However, they do hold a significant part of the BBC's institutional memory. They can hold content tightly related to current news events, which could be extremely valuable to help contextualise those events. Most of the programmes within the BBC World Service archive, for example, have not been listened to since they were originally broadcast, and they cover a very wide range of topics over a number of decades.

In this paper we describe a system that enables content from uncatalogued parts of the BBC's archive to be surfaced alongside current news events. In particular, we focus on the BBC World Service archive and how this work is made possible by a combination of Semantic Web technologies, automated interlinking, user feedback and data visualisation.

Our system starts by automatically deriving links from archive content to Linked Data URIs. We use the resulting data to publish the archive and bootstrap search and discovery within it. We then let users validate, correct and augment these automatically derived links. As a result of this feedback, the interlinks between our archive and the Semantic Web are continuously improving. We also automatically extract topics from live BBC News subtitles. The resulting interlinks between live news subtitles and the BBC's archive are used in a dynamic visualisation enabling journalists and editors to quickly locate relevant archive content. This content can then be used to provide historical context, background information and supporting content around current affairs. An architectural overview of our system is available in Figure 1.

The paper is organised as follows. In Section 2 we briefly describe various efforts aiming at cataloguing the BBC archive. We then describe the BBC World Service archive. In Section 3 we describe our automated tools for interlinking archive content with the Semantic Web. In Section 4 we describe how such automatically derived links are being used to publish this archive content online, and the mechanisms we put in place to enable people to feed back on the quality of those links. Finally in Section 5 we describe how we use these interlinks and topic extraction from live news subtitles to find and visualise archive content related to current news events.

2 Background

2.1 Cataloguing the archive

A number of cataloguing efforts have been made to improve the ease with which people can find content in the BBC archive. This cataloguing effort has been geared towards reuse. In other words to enable programme makers to easily find clips of content to include in their own, newly commissioned, programmes. The coverage of the catalogue is not uniform across the BBC's archive, for example it excludes the BBC World Service, which has been broadcasting since 1932. Creating this metadata is a time and resource expensive process; a detailed

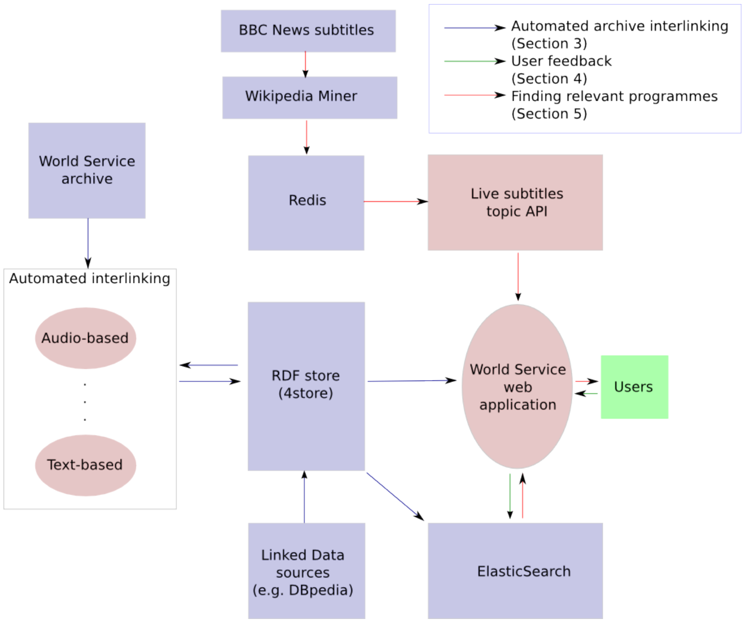

BBC News subtitles
Wikipedia Miner
Automated archive interlinking (Section 3)
User feedback (Section 4)
Finding relevant programmes (Section 5)
World Service archive
Redis
Live subtitles topic API
Automated interlinking
Audio-based
Text-based
RDF store (4store)
World Service web application
Users
Linked Data sources (e.g. DBpedia)
ElasticSearch

Fig. 1. Overview of the architecture of a system for finding archive content related to current news events.

analysis of a 30 minute programme can take a professional archivist 8 to 9 hours. Moreover, as this data is geared towards professional reuse, it is often not appropriate for driving user-facing systems — it is either too shallow (not all programmes are being classified) or too deep (information about individual shots or rushes).

There have been a number of attempts at trying to automatically classify the BBC archive. The THISL system [1] applied an automated speech recognition system (ABBOT) on BBC news broadcasts and used a bag-of-words model on the resulting transcripts for programme retrieval. The Rich News system [7] also used ABBOT for speech recognition. It then segmented the transcripts using bag-of-words similarity between consecutive segments using Choi's C99 algorithm [6]. For each segment a set of keyphrases was extracted and used, along with the broadcast date of the programme, to find content within the BBC News web site. Information associated with retrieved news articles was then used to annotate the topical segment. Recent work at the BBC classifies archived programmes according to their mood [8] and investigates ways for users to use mood as a way to explore the archive.

2.2 Tagging with Linked Data URIs

Since 2009 the places, people, subjects or organisations mentioned in new programmes have been “tagged” with DBpedia [2] URIs, effectively interlinking these programmes with the Semantic Web. These tags allow the BBC’s audience to easily find programmes relating to particular topics, by presenting them through a navigable web interface at `bbc.co.uk/programmes`. These tags are also being used to drive topic-based navigation within published parts of the BBC’s archive, such as the In Our Time archive[1].

The tool used by editors to tag programmes suggests tags based on supporting textual metadata, for example a synopsis or a title. Additional tags are then manually associated with the programme. The entire tagging process is described in more detail in [9].

A benefit of using Linked Data[2] URIs as tags is that they are unambiguous and that we can retrieve more information about those tags when needed. For example, programmes tagged with places can be plotted on a map, or topic-based aggregation pages can be enriched with information about the corresponding topic. By having these anchor points in the Linked Data web, we can accommodate a wide range of unforeseen use-cases.

This process of manual tagging is naturally very time-consuming, and with the emphasis on delivering new content, would take considerable time to apply to the entire archive. This problem is compounded by the lack of availability of textual metadata for a significant percentage of the archive which prevents the bootstrapping of the tagging process.

2.3 The BBC World Service archive

The BBC World Service was until last year operated by the BBC on behalf of the UK government so had its own archiving system and process. It was therefore excluded from the cataloguing efforts mentioned previously. This archive consists of digitised copies of all tapes that have been saved of prerecorded programmes broadcast on the English language part of the World Service since 1947. It currently holds around 50,000 programmes with associated audio files. This amounts to about three years of continuous audio and around 15TB of data.

However the metadata around this archive is relatively sparse and sometimes wrong. In the best cases it includes a series title (e.g. ‘From Our Own Correspondent’ although those titles are often not consistently spelled), approximate broadcast date (although a hundred programmes are reporting a broadcast date in the future or before the start of the World Service), a title (19,000 programmes have no titles) and a synopsis (17,000 programmes have an empty synopsis).

On a more positive note, the full audio content is available in digital form. We therefore consider bootstrapping search and discovery within this archive by exploiting the audio content itself as well as textual metadata when it is present.

[1] See `http://www.bbc.co.uk/programmes/b006qykl`.

[2] See `http://linkeddata.org`

In the rest of this paper we focus on the BBC World Service archive, as an example of an uncatalogued part of the BBC's archive.

3 Automated archive interlinking

It would take a significant amount of time and resource to manually annotate the parts of the BBC archive with little or no metadata. We therefore consider bootstrapping this annotation process using a suite of automated interlinking tools working from text and from audio.

3.1 Topics from textual metadata

In some cases, textual metadata is available alongside archive content. In the case of the BBC World Service archive, this data could be a synopsis or a title for the programme. In other cases, it could be a script, production notes, etc. We consider using this data when it is available to try and associate the programme with a number of topics identified by Linked Data URIs.

We process this textual data using an instance of Wikipedia Miner [11]. Wikipedia Miner learns from the structure of links between Wikipedia pages and uses the resulting model to provide a service detecting potential Wikipedia links in unstructured text. We trained a Wikipedia Miner instance with a Wikipedia dump from August 2012. Wikipedia Miner returns a set of Wikipedia identifiers for the various topics detected in the text, which we then map to Linked Data identifiers using the DBpedia Lite[3] service. Each of these topics is also associated with a confidence score. We store the resulting weighted associations between programmes and topics in a shared RDF store[4]. For the whole archive, this process generated around 1 million RDF triples, interlinking this archive with DBpedia.

3.2 Topics from audio

We also consider using the content itself to identify topics for these programmes. This is motivated by the fact that a lot of these programmes will have very little or no associated textual metadata. Where textual metadata is present it will rarely tackle all the topics discussed within the programme.

The full description of this algorithm to extract topics from audio as well as its evaluation is available in [13]. The core algorithm and our evaluation dataset are available on our Github account[5].

We start by identifying the speech parts within the audio content. An implementation of the algorithm for speech–music segmentation described in [15] is

[3] See `http://dbpedialite.org/`.

[4] We use 4store, available at `http://4store.org`.

[5] See `https://github.com/bbcrd/rdfsim` for the algorithm and `https://github.com/bbcrd/automated-audio-tagging-evaluation` for the evaluation dataset and a script which can be used to reproduce our evaluation results.

available as a Vamp plugin [5] on our Github account[6]. We then automatically transcribe the speech parts. We use the open source CMU Sphinx-3 software, with the HUB4 acoustic model [16] and a language model extracted from the Gigaword corpus. The resulting transcripts are very noisy. We evaluated the average Word Error Rate on the BBC Reith Lectures, a publicly available dataset of transcribed programmes covering almost each year since 1976 and a wide range of different speakers. We got an average Word Error Rate of around 55%[7]. Most off-the-shelf concept tagging tools perform badly on noisy automated transcripts as they rely on the input text to be hand-written and to include clues such as capitalisation and punctuation which our transcripts are lacking. We therefore designed an alternative concept tagging algorithm which does not assume any particular structure in the input text.

We start by generating a list of URIs used by BBC editors to tag programmes as part of the process described in Section 2.2. Those URIs identify people, places, subjects and organisations within DBpedia. This list of identifiers constitutes our target vocabulary. We dereference these identifiers and get their labels from their `rdfs:label`[8] property. We strip out any disambiguation string from the label and apply the Porter Stemmer algorithm [12]. We apply the same stemming algorithm to the automated transcripts and look for those stemmed labels within them. The output of this process is a list of candidate terms found in the transcripts and a list of possible corresponding DBpedia URIs for them. For example if 'london' was found in the transcripts it could correspond to at least two possible DBpedia URIs: `d:London` and `d:London,_Ontario`. Our algorithm uses the structure of DBpedia itself to disambiguate and rank these candidate terms, and in particular a similarity measure capturing how close two URIs are from each other in the DBpedia graph. For example if the automated transcripts mention 'london', and 'england' a lot, our algorithm will pick `d:London` as the correct disambiguation for the former, as it is very close to one possible disambiguation of the latter, i.e. `d:England`. We end up with a ranked list of DBpedia URIs for each programme. Some examples of the top three tags and their associated scores are given in Table 1 for three different programmes.

We evaluated our algorithm on a dataset of 132 programmes with manual tagging data added through the process described in Section 2.2 and made available as part of the `bbc.co.uk/programmes` Linked Data [14]. We use the `TopN` measure introduced by Berenzweig et al. in [3] for the evaluation of automated music tagging algorithms.

$$\mathtt{TopN} = \frac{\sum_{\mathtt{j}=1}^{\mathtt{N}} \alpha_{\mathtt{c}}^{\mathtt{k_j}}}{\sum_{\mathtt{i}=1}^{\mathtt{N}} \alpha_{\mathtt{c}}^{\mathtt{i}}}$$

[6] See `https://github.com/bbcrd/bbc-vamp-plugins`.

[7] The dataset and an evaluation script to reproduce this result are available at `https://github.com/bbcrd/bbc-reith-lectures-sphinx-evaluation`.

[8] We use the namespaces defined at the end of the paper.

N is the number of tags available in bbc.co.uk/programmes and k_j is the position of tag j in the automatically extracted tags. α_c is an exponential decay constant which we set at 0.8, expressing how much we want to penalise a tag for appearing down the list of automated tags.

A baseline random tagger gives a TopN measure of 0.0002. Our algorithm gives 0.205, and the best third-party concept tagging algorithm we evaluated, using the automated transcripts as an input and TF-IDF ranking, gives 0.1951. We are currently working with different projects, such as DBpedia Spotlight [10], to try and improve the results of automated concept tagging algorithms on noisy automated transcripts.

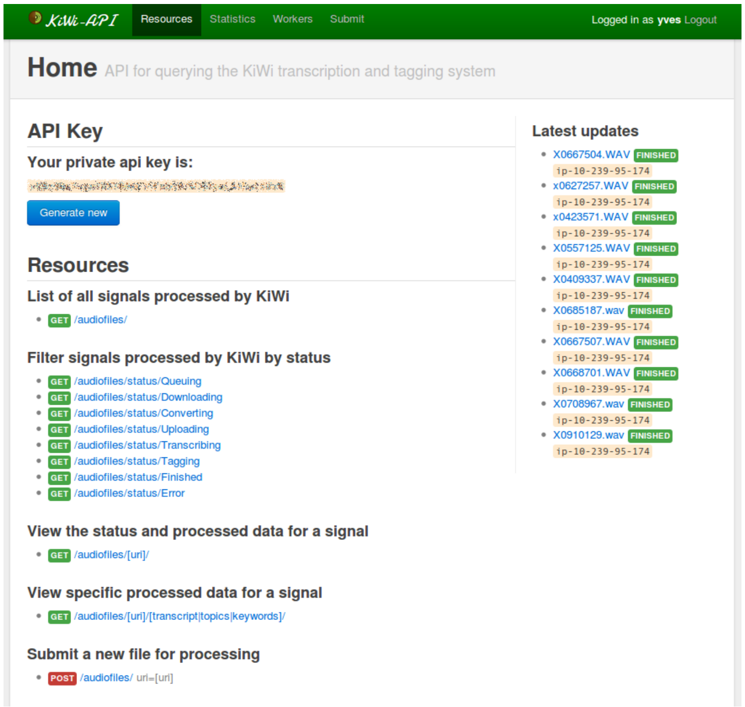
The algorithm works well for programmes that only deal with a few topics but its performance decreases as the number of topics mentioned in the programme increases. For example it performs very well on documentaries and factual programmes but performs poorly on magazine programmes. On the latter type of programmes, our algorithm will struggle to find a clear disambiguation for candidate terms and a clear set of representative topics. A way to mitigate this issue is to start our tagging process with a topic segmentation of the programme, which we are currently investigating. It would also ensure we can find the relevant segment of a programme when researching a specific topic, rather than pointing to either the whole programme or specific timestamps at which the topic is mentioned.

Tag	Score
Programme 1	
d:Benjamin_Britten	0.09
d:Music	0.054
d:Gustav_Holst	0.024
Programme 2	
d:Revolution	0.037
d:Tehran	0.032
d:Ayatollah	0.025
Programme 3	
d:Hepatitis	0.288
d:Vaccine	0.129
d:Medical_research	0.04

Table 1. Example of automatically generated tags and associated scores. Programme 1 is a 1970 profile of the composer Gustav Holst. Programme 2 is a 1983 profile of the Ayatollah Khomeini. Programme 3 is a 1983 episode of the Medical Programme.

3.3 Automated interlinking for large audio archives

It would take around 4 years to transcribe the entire World Service archive on commodity hardware. We therefore developed an infrastructure to process entire radio archives in a reasonable time. We separated each step of the workflow into independent, self-contained applications, or "workers". Each worker takes input in the form of the results of the previous step of the workflow, and produces output to be given to the next step of the workflow. These workers will decode and downsample programmes, upload the resulting data to shared cloud storage, transcribe the programmes, and extract and rank tags from the resulting transcripts. We also configured a message-queuing system using RabbitMQ[9] to allow workers to pick up new tasks and assign tasks to one-another. In order to control and monitor the system as well as centralise the resulting data, we developed an HTTP interface called "KiWi API" which has direct access to the message-queuing system. A capture of the homepage of KiWi API is given in Figure 2.



Home API for querying the KiWi transcription and tagging system

API Key

Your private api key is:

[redacted]

Resources

List of all signals processed by KiWi

- GET /audiofiles/

Filter signals processed by KiWi by status

- GET /audiofiles/status/Queuing
- GET /audiofiles/status/Downloading
- GET /audiofiles/status/Converting
- GET /audiofiles/status/Uploading
- GET /audiofiles/status/Transcribing
- GET /audiofiles/status/Tagging
- GET /audiofiles/status/Finished
- GET /audiofiles/status/Error

View the status and processed data for a signal

- GET /audiofiles/[uri]/

View specific processed data for a signal

- GET /audiofiles/[uri]/[transcript|topics|keywords]/

Submit a new file for processing

- POST /audiofiles/ uri=[uri]

Latest updates

- X0667504.WAV FINISHED ip-10-239-95-174
- x0627257.WAV FINISHED ip-10-239-95-174
- x0423571.WAV FINISHED ip-10-239-95-174
- X0557125.WAV FINISHED ip-10-239-95-174
- X0409337.WAV FINISHED ip-10-239-95-174
- X0685187.wav FINISHED ip-10-239-95-174
- X0667507.WAV FINISHED ip-10-239-95-174
- X0668701.WAV FINISHED ip-10-239-95-174
- X0708967.wav FINISHED ip-10-239-95-174
- X0910129.wav FINISHED ip-10-239-95-174

Fig. 2. The home page of KiWi API

We then built an Amazon Machine Image (AMI[10]) with those workers pre-installed. This AMI can be deployed on a large number of instances and automatically spawns a number of workers when starting up, depending on the

[9] See `http://www.rabbitmq.com/`.
[10] See `https://aws.amazon.com/amis/`

number of CPUs and the amount of memory available. With this infrastructure in place, we processed the entire BBC World Service archive in two weeks instead of years for a pre-determined cost and generated a collection of ranked Linked Data tags for each BBC World Service programme. For the whole archive, the automated audio interlinking generated around 5 million RDF triples, interlinking this archive with DBpedia and the rest of the Linked Data cloud. We are currently using this same API to process content from other archives within the BBC. The only bottleneck in how quickly an entire archive can be processed is the bandwidth between our content servers and cloud storage servers.

4 Validation of automated links

We now have an automated set of links for each programme, which we can use to bootstrap search and navigation within the BBC World Service archive. Topic data can be used for browsing between programmes, generating topic-based aggregations and searching for programmes on specific topics. We built an application using these links to publish this archive on the web[11].

This web site is built using the data held within our shared RDF store. This store includes the automated interlinks mentioned above as well as all the data we could gather around this archive. It also includes a set of images extracted from Ookaboo[12] which we use to generate programme depictions from the list of topics associated with them. Overall, we store around 20 million RDF triples. Most pages are built from SPARQL queries issued to that store with an average response time of 15ms.

Automated data will never be entirely accurate so mechanisms are in place for registered users to correct data when it is found to be wrong. When logged in, users can upvote or downvote each individual topic and can add new topics through an auto-completed list, using DBpedia as a target vocabulary. A screenshot of the interface for a 'Discovery' programme on smallpox[13] is available in Figure 3.

The aggregate of positive and negative votes on each tag is used to improve the machine-generated ranking, and will have an impact on which programmes will be retrieved when a user searches for a particular topic. Gathering this user feedback makes it possible to automatically refine the automated algorithms. This in turns leads to better automated metadata for the rest of the archive creating a useful feedback cycle that leads to a better and better archive experience. As a result of this feedback cycle, the interlinks between our archive and the Semantic Web are continuously improving.

The web site launched in late August 2012 and we are progressively increasing the number of registered users. We now have around 2,000 users. As of April 2013 we have had more than 40,000 positive and negative votes against automatically

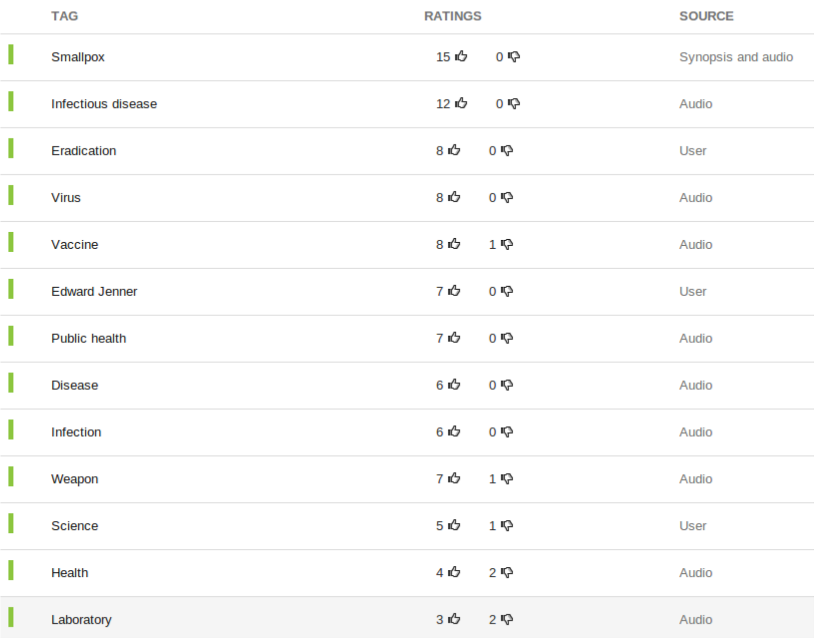
[11] See `http://worldservice.prototyping.bbc.co.uk`.

[12] See `http://ookaboo.com/`

[13] See `http://worldservice.prototyping.bbc.co.uk/programmes/X0909348`.

generated topics, covering around 6,000 distinct programmes. Around 10,000 new topics were added by users to around 3,000 distinct programmes.

As well as refining search and discovery within the archive and helping us improve our algorithm, this user data is also helping us to continuously evaluate our automated interlinking results. The raw user data can be used to evaluate how well our algorithm is doing and we are also tracking the progress of the evaluation measure mentioned above to see how the quality of our interlinks is evolving. A detailed report on the evolution of overall interlinking quality remains future work.



	TAG	RATINGS		SOURCE
▌	Smallpox	15 👍	0 👎	Synopsis and audio
▌	Infectious disease	12 👍	0 👎	Audio
▌	Eradication	8 👍	0 👎	User
▌	Virus	8 👍	0 👎	Audio
▌	Vaccine	8 👍	1 👎	Audio
▌	Edward Jenner	7 👍	0 👎	User
▌	Public health	7 👍	0 👎	Audio
▌	Disease	6 👍	0 👎	Audio
▌	Infection	6 👍	0 👎	Audio
▌	Weapon	7 👍	1 👎	Audio
▌	Science	5 👍	1 👎	User
▌	Health	4 👍	2 👎	Audio
▌	Laboratory	3 👍	2 👎	Audio

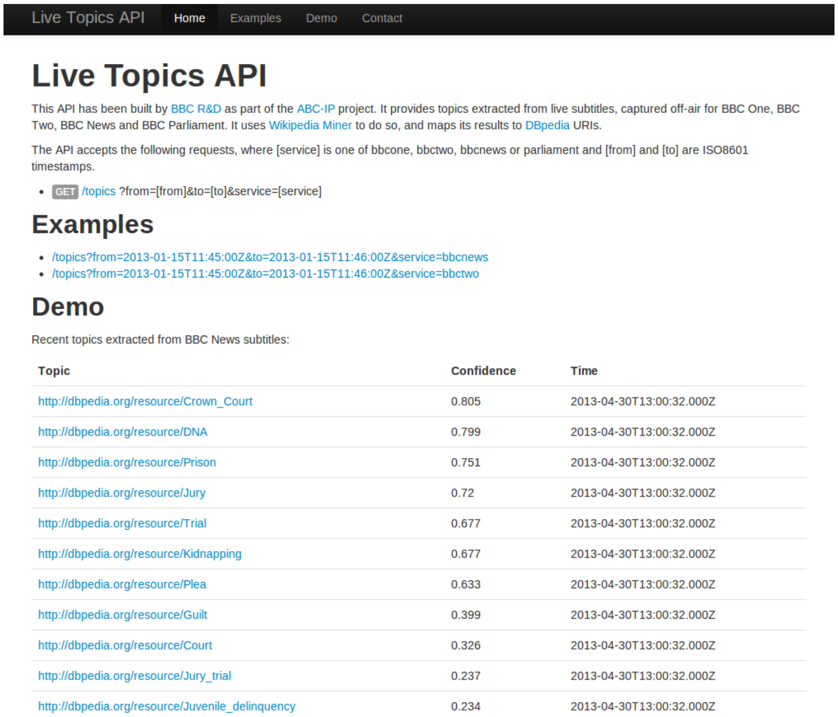
Fig. 3. A set of topics along with their origin and associated user validation data around a ‘Discovery’ programme on smallpox. Topics can be derived from textual metadata (‘synopsis’), audio or can be added by users. When logged in, users can upvote or downvote individual tags by clicking on the thumbs button.

We index the resulting topic and voting data against programmes in an ElasticSearch instance[14] in order to perform fast topic-based searches across the entire archive. This index takes into account all this user feedback as well as the weights assigned to individual topics by our automated tagging algorithm. We use this index to drive an archive-wide search, but also to quickly surface content related to current news events, as detailed in the next section.

[14] See `http://www.elasticsearch.org/`.

5 Finding archive programmes related to current news events

Another part of our system is to automatically detect which topics are being discussed around current news events. In order to do this we capture the live subtitles for the BBC News TV channel. The subtitles are then aggregated during consecutive 40 second intervals. We process those 40 seconds of subtitles with the same Wikipedia Miner setup mentioned in Section 3.1. We store the resulting time-stamped topics in a Redis instance[15] providing a publish/subscribe mechanism for live topics. We also built a simple HTTP API to access topics mentioned at a particular time on a particular channel. A screenshot of that API is available in Figure 4.



Live Topics API

This API has been built by BBC R&D as part of the ABC-IP project. It provides topics extracted from live subtitles, captured off-air for BBC One, BBC Two, BBC News and BBC Parliament. It uses Wikipedia Miner to do so, and maps its results to DBpedia URIs.

The API accepts the following requests, where [service] is one of bbcone, bbctwo, bbcnews or parliament and [from] and [to] are ISO8601 timestamps.

- GET /topics ?from=[from]&to=[to]&service=[service]

Examples

- /topics?from=2013-01-15T11:45:00Z&to=2013-01-15T11:46:00Z&service=bbcnews
- /topics?from=2013-01-15T11:45:00Z&to=2013-01-15T11:46:00Z&service=bbctwo

Demo

Recent topics extracted from BBC News subtitles:

Topic	Confidence	Time
http://dbpedia.org/resource/Crown_Court	0.805	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/DNA	0.799	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Prison	0.751	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Jury	0.72	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Trial	0.677	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Kidnapping	0.677	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Plea	0.633	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Guilt	0.399	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Court	0.326	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Jury_trial	0.237	2013-04-30T13:00:32.000Z
http://dbpedia.org/resource/Juvenile_delinquency	0.234	2013-04-30T13:00:32.000Z

Fig. 4. The Live Topics API, showing time-stamped topics extracted from the BBC News channel.

We now have a stream of anchor points within the Linked Data cloud, identifying which topics are being discussed on the BBC News channel. We also have

[15] See `http://redis.io/`.

an index of archive programmes against Linked Data URIs that is continuously being updated and refined. By using the interlinks between these two datasets, we can find archive programmes related to current news events.

In order to do this we query our ElasticSearch index for programmes matching 'current' topics, or topics that were mentioned on the BBC News channel in the last five minutes. Programmes matching those topics will be returned, with a given weight taking into account automated weights and user voting data. We further refine those weights by taking into account the number of current topics those programmes also match. The more current topics a programme matches, the more likely it is to be related to a current news event.

The resulting data is made available through the World Service archive prototype described earlier and used to drive a dynamic client-side visualisation[16]. This visualisation is depicted in Figure 5 and in Figure 6. The blue dots are topics mentioned in the last five minutes on the BBC News channel. The size of these blue dots is driven by the weights returned by Wikipedia Miner. Each small dot is an archive programme related to those topics. The redder a dot is, the more relevant the programme is. This visualisation is based on D3.js[17] and dynamically updates as new topics get mentioned on BBC News. Hovering over the programme stabilises the visualisation around that programme and provides more information as well as a link to the programme.

This visualisation shows archive content related to current news events. This archive content can then be used to provide some context around a particular event. For example, a recent news event about replacing poppy cultivation by cotton in Afghanistan was represented by the topics 'Opium poppy', 'Afghanistan' and 'Cotton' in the Live Topics API. The visualisation picked up a 2008 programme about a new opium ban in Afghanistan and the impact it had on local farmers. Another recent news event about a measles outbreak lead to two programmes being highlighted by the visualisation: a 1983 'Medical Programme' episode on techniques for measles immunisation and a 2000 'Science in Action' episode on predicting measles outbreaks.

6 Conclusions and future work

In this paper we have described a system for finding archive programmes related to current news events. These archive programmes can be used to provide historical context, background information and supporting content around particular events. We specifically focused on parts of the archive that have little or no pre-existing metadata as very little reuse is currently made of them.

This system is driven by interlinking both archive content and live subtitles with the Semantic Web. For archive content we use automated interlinking techniques from supporting textual metadata and audio content. This results in a set of topics identified by their DBpedia URIs for each programme in the archive.

[16] See `http://worldservice.prototyping.bbc.co.uk/visualisations/current`. Access to the visualisation requires registration for the time being.

[17] See `http://d3js.org/`.

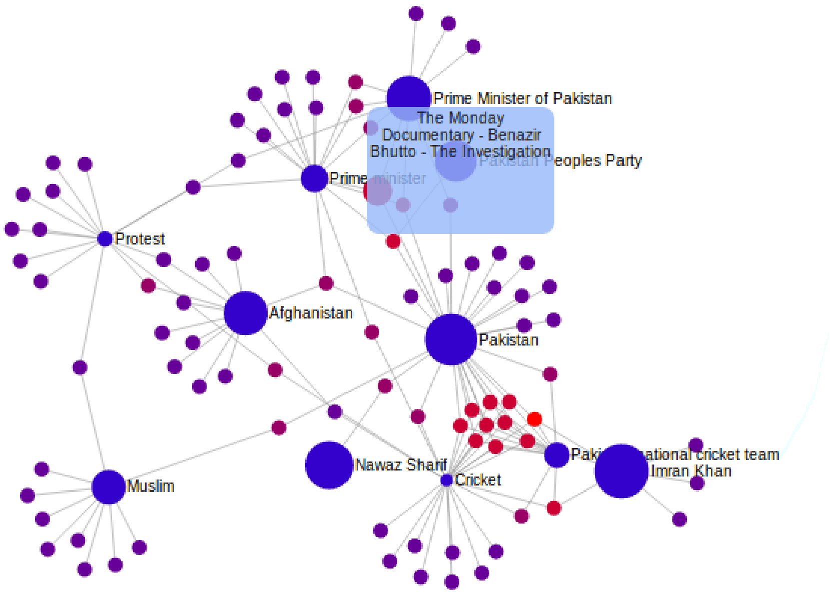


Fig. 5. Visualising archive programmes related to current news events. This capture of the visualisation was taken during the May 2013 Prime Ministerial election in Pakistan (involving Imran Khan, a politician and former cricketer) was discussed on the news. The red programmes in this visualisation include a 1990 Benazir Bhutto documentary and a 2003 Imran Khan interview.

We then use these interlinks to drive a web application enabling users to navigate the archive and validate, correct and augment those links. This results in a continuously improving set of interlinks between our archive content and DBpedia. We also automatically extract topics identified by DBpedia URIs from live BBC News subtitles. The resulting interlinks between live news subtitles and archive content are then used in a dynamic visualisation, showing programmes related to current news events. The visualisation also shows how likely programmes are to be related to current news events, enabling journalists or editors to quickly locate relevant archive content. This archive content can then be used to provide more context around particular events.

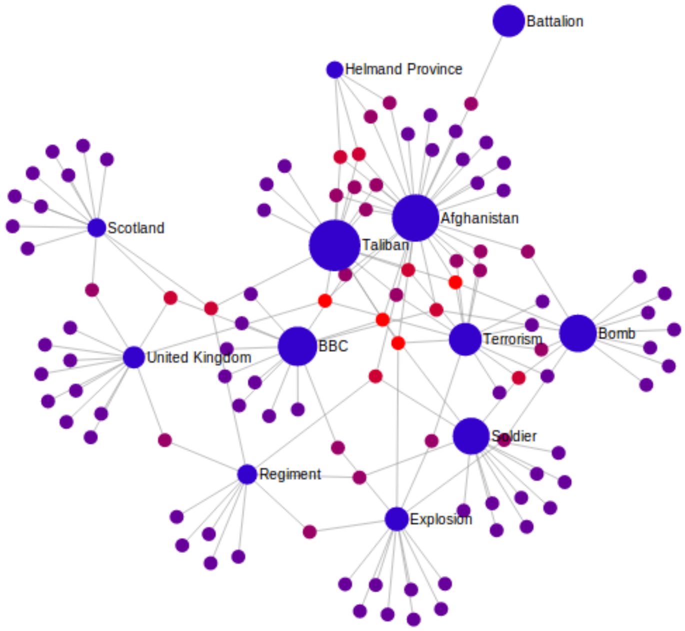


Fig. 6. Another visualisation, this time generated by a news story about UK soldiers in Afghanistan. One of the programmes brought up by this visualisation is a 2008 programme from a BBC correspondent in the Helmand Province, describing the evolution of the region over a year.

We are currently tracking the evolution of interlinking quality for the World Service archive as we accumulate more and more user feedback. A detailed report on this evolution remains future work. We also recently developed an algorithm to quickly identify contributors in and across programmes, using speaker supervectors [4] and an index based on Locality-Sensitive Hashing [17][18]. We are currently working on ways of interlinking these contributors with other datasets using a similar mixture of automation and crowdsourcing. These interlinks would enable us to surface programmes featuring people mentioned in the news in this visualisation. For example interviews from the archive featuring particular politicians could be surfaced alongside news events involving them. We also want to investigate grouping topics into actual events, e.g. 'Measles', 'Outbreak' and 'Swansea' could be grouped into a single event as defined in the Storyline ontology[19]. The time-stamped topics data we get from live subtitles would be very useful for that. This would enable more precise event-based discovery within the archive. We are also working on automated programme segmentation. Some programmes are fairly long and tackle multiple topics which has a negative impact on our automated interlinking algorithm and on the reusability of archive programmes found by our visualisation. Finally, we recently started work on a platform for sharing our automated interlinking tools and cloud-based processing framework with other content owners outside of the BBC.

Acknowledgements

The research for this paper was conducted as part of the Automatic Broadcast Content Interlinking Project (ABC-IP). ABC-IP is a collaborative research and development initiative between the British Broadcasting Corporation and MetaBroadcast Ltd, supported with grant funding from the UK Technology Strategy Board under its 'Metadata: increasing the value of digital content (mainstream projects)' competition from September 2010.

Annex: Namespaces

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix d: <http://dbpedia.org/resource/> .
@prefix c: <http://dbpedia.org/resource/Category:> .
```

References

1. Dave Abberley, David Kirby, Steve Renals, and Tony Robinson. The THISL broadcast news retrieval system. In *Proc. ESCA Workshop on Accessing Information In Spoken Audio*, 1999.
2. S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the International Semantic Web Conference*, Busan, Korea, November 11-15 2007.

[18] See `https://github.com/bbcrd/ruby-lsh` for our implementation of Locality-Sensitive Hashing and `http://worldservice.prototyping.bbc.co.uk/programmes/X0403940` for an example of how the resulting data is currently being used.

[19] See `http://www.bbc.co.uk/ontologies/storyline/2013-05-01.html`.

3. Adam Berenzweig, Beth Logan, Daniel P. W. Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, Summer 2004.
4. W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using gmm supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311, May 2006.
5. Chris Cannam, Christian Landone, Mark Sandler, and Juan Pablo Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the International Conference on Music Information Retrieval*, 2006.
6. Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, 2000.
7. Mike Dowman, Valentin Tablan, Hamish Cunningham, and Borislav Popov. Web-assisted annotation, semantic indexing and search of television and radio news. In *WWW '05 Proceedings of the 14th international conference on World Wide Web*, 2005.
8. J. Eggink and D. Bland. A large scale experiment for mood-based classification of tv programmes. In *Proc. IEEE Int. Conf. on Multimedia and Expo, ICME2012*, July 2012.
9. Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Chris Bizer, and Robert Lee. Media meets semantic web - how the BBC uses DBpedia and linked data to make connections. In *Proceedings of the European Semantic Web Conference In-Use track*, 2009.
10. P. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.
11. David Milne and Ian H. Witten. Learning to link with wikipedia. In *CIKM proceedings*, 2008.
12. M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130137, 1980.
13. Yves Raimond and Chris Lowis. Automated interlinking of speech radio archives. In *Proceedings of the Linked Data on the Web workshop, World Wide Web conference*, 2012.
14. Yves Raimond, Tom Scott, Silver Oliver, Patrick Sinclair, and Michael Smethurst. *Linking Enterprise Data*, chapter Use of Semantic Web technologies on the BBC Web Sites, pages 263–283. Springer, 2010.
15. J. Saunders. Real-time discrimination of broadcast speech/music. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996.
16. Kristie Seymore, Stanley Chen, Sam-Joo Doh, Maxine Eskenaziand Evandro Gouvea, Bhiksha Raj, Mosur Ravishankar, Ronald Rosenfeld, Matthew Siegler, Richard Sternane, and Eric Thayer. The 1997 CMU sphinx-3 english broadcast news transcription system. In *Proceedings of the DARPA Speech Recognition Workshop*, 1998.
17. Malcolm Slaney and Michael Casey. Locality-sensitive hashing for finding nearest neighbors. *IEEE Signal Processing Magazine*, pages 128–131, March 2008.