Automated interlinking of speech radio archives

Yves Raimond, Chris Lowis
BBC R&D
London, United Kingdom
{yves.raimond,chris.lowis}@bbc.co.uk

ABSTRACT

The BBC is currently tagging programmes manually, using DBpedia as a source of tag identifiers, and a list of suggested tags extracted from their synopsis. These tags are then used to help navigation and topic-based search of BBC programmes. However, given the very large number of programmes available in the archive, most of them having very little metadata attached to them, we need a way of automatically assigning tags to programmes. We describe a framework to do so, using speech recognition, text processing and concept tagging techniques. We evaluate this framework against manually applied tags, and compare it with related work. We find that this framework is good enough to bootstrap the interlinking process of archived content.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Linked Data,Concept Tagging,Speech Processing

1. INTRODUCTION

The BBC (British Broadcasting Corporation) has broadcast radio programmes since 1922. Over the years, it has accumulated a very large archive of programmes. A number of cataloguing efforts have been made to improve the ease with which people can find content in this archive. This cataloguing effort has been geared towards reuse, in other words to enable programme makers to easily find snippets 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 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).

Since 2009 the places, people, subjects or organisations mentioned in new programmes have been "tagged" with DBpedia [2] web identifiers. These tags allow the BBC's audience to easily find programmes relating to particular topics, by presenting them through a navigable web interface at `http://bbc.co.uk/programmes`. The tool used by editors to tag programmes suggests tags based on the textual content, for example a synopsis, or title, associated with a programme. Tags are then manually associated with the programme. The entire tagging process is described in more details in [8]. A benefit of using Linked Data web identifiers 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 plot on a map, or 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 meta-data for a significant percentage of the archive which prevents the bootstrapping of the tagging process.

On a more positive note, the full audio content is, in the case of the World Service radio archive, available in digital form. The archive currently holds around 70,000 programmes, which amounts to about two and a half years of continuous audio. In this paper, we describe a framework to automatically interlink such an archive with the Linked Data Web, by automatically tagging individual programmes with Linked Data web identifiers.

We start by describing related work. We then describe a novel approach which uses an open-source speech recognition engine, and how we process the transcripts it generates to extract relevant tags that can be used to annotate the corresponding radio programme. We evaluate the results by comparing the tags generated by this method with those manually applied by editors to BBC programmes. We compare the results of our method with those obtained by other, existing methods.

LDOW2012, April 16, 2012, Lyon, France

2. RELATED WORK

This paper is concerned with two topics: the classification of the BBC archive and, more generally, the problem of automatically applying semantic labels to a piece of recorded audio.

There has been a number of attempts at trying to automatically classify the BBC archive. The THISL system [1] applies an automated speech recognition system (ABBOT) on BBC news broadcasts and uses a bag-of-words model on the resulting transcripts for programme retrieval. The Rich News system [7] also uses ABBOT for speech recognition. It then segments the transcripts using bag-of-words similarity between consecutive segments using Choi's C99 algorithm [5]. For each segment, a set of keyphrases is extracted and used, along with the broadcast date of the programme, to find content within the BBC News website. Information associated with retrieved news articles is then used to annotate the topical segment. Recent work at the BBC classifies the mood of archived programmes using their theme tunes [6] and ultimately intends to help users browse the archive by mood.

Several researchers have tried to automatically reproduce the labelling task of a piece of speech audio. The first work in that area [23] details a supervised automated classification method which can assign a particular piece of audio to one of six topic classes. Paaß et al. [18] describe a classifier that can assign speech audio to genre topics drawn from the Media Topic taxonomy of the International Press Telecommunications Council[1]. Makhoul et al. [10] describe a well-integrated system of technologies for indexing and information retrieval on automatically transcribed audio. Their topic assignment algorithm is a probabilistic Hidden Markov Model whose parameters are trained on a corpus of existing documents with human assigned topic labels. Olsson and Oard describe techniques for assigning topic labels to automated transcripts [17]. Here the topic labels are drawn from the CLEF CL-SR[2] English oral history thesaurus. Their techniques leverage temporal aspects of the target audio such as the fact they typically have a chronological narrative. This means that labels can be assigned with a greater probability based on the co-occurrence within the transcript.

In comparison to the technique presented in this paper the Olsson and Oard and Makhoul methods are supervised. They require the models to be trained on an existing set of transcripts and their corresponding topics as assigned by a human indexer. The technique presented here attempts topic classification in an unsupervised manner using the automated transcript alone, as we will see later.

There is a significant corpus of work on discovering the main topics of textual documents. A number of possible approaches have been investigated, including:

- Probabilistic topic models, e.g. [4], where documents are modelled as being drawn from a finite mixture of underlying topics;
- Term assignment, e.g. [11], where the Medical Subject Heading (MeSH) vocabulary is used as a controlled vocabulary, and a classifier is trained to associate documents with terms in that vocabulary;
- Keyphrase extraction, e.g. [26], where a classifier is trained to assign probabilities to possible keyphrases;
- Automated tagging, e.g. [15], where similar and already tagged documents are found, and used as a basis for suggesting tags.

The work that is most related to ours is [12], where Wikipedia web identifiers are used as tags and automatically assigned to textual documents. A 'keyphraseness' measure is first used to identify words that are likely to be specific to the topics expressed in the document. Each candidate is then associated with a Wikipedia article capturing its meaning. We use a similar workflow, but introduce a new automated tagging algorithm based on structured data available as Linked Data, and suitable for automatically generated transcripts, which can be very noisy.

3. BACKGROUND

In order to find appropriate tags to apply to programmes within the archive, we build on top of the Enhanced Topic-based Vector Space Model proposed in [9] and further described and evaluated in [19]. We describe this model in this Section.

3.1 Vector Space Model

First, we define a couple of concepts:

- *term* — a symbol, e.g. 'cat' or 'house';
- *document* — an ordered set of terms;
- *corpus* — a set of documents.

We then consider a vector per document $\vec{d}$, where each dimension corresponds to a term t with a weighting $w_{d,t}$. TF-IDF proved a very popular way of deriving those weights, and includes both local (the TF is relevant to the document) and global (the IDF is relevant to the corpus) factors. Document similarity can then be captured by the cosine similarity between the two document vectors.

$$\cos(\vec{d_i}, \vec{d_j}) = \frac{\vec{d_i}\vec{d_j}}{\|\vec{d_i}\|\|\vec{d_j}\|}$$

3.2 Topic-based Vector Space Model

A Topic-based Vector Space Model (TVSM) considers documents as vectors in a vector space, in which all dimensions are so-called fundamental topics, which are defined as being inter-independent. We consider a vector $\vec{t}$ in that space for each term. The normalised and weighted sum of all the term vectors in a document gives us a document vector $\vec{d}$.

$$\vec{d} = \frac{1}{\| \sum w_{d,t}\vec{t}\|} \sum w_{d,t}\vec{t}$$

As above, we consider the similarity between two documents as being the cosine similarity between the two document vectors. We can compute this similarity by knowing the length of the term vectors $\vec{t}$ and their angles between one another. TVSM does not specify an approach for obtaining those lengths and angles.

[1] See http://www.iptc.org, last accessed November 2011

[2] Cross-Language Evaluation Forum Cross-Language Speech Retrieval track

3.3 Enhanced Topic-based Vector Space Model

An Enhanced Topic-based Vector Space Model (eTVSM) embeds ontological information in TVSM, by obtaining document similarities not by using similarities between terms, but by using mappings from those terms to an ontological space — a vector space capturing the structure of an ontology.

This is particularly relevant for us, as we want to tag programmes with web identifiers, which can themselves link to various web ontologies. For example, DBpedia web identifiers link to the DBpedia ontology, to a SKOS categorisation system [14] derived from the Wikipedia categories, and to the YAGO ontology [25]. In the following, we formalise eTVSM in such a context.

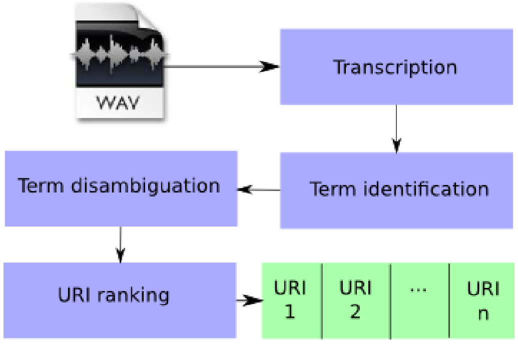

Figure 1: Workflow of our automated tagging process

We define two new concepts:

- *interpretation* — a particular term can have multiple interpretations. For example, the term `bar` has at least two interpretations: `d:Bar_(music)` and `d:Bar_(unit)`[3];
- *category* — a particular interpretation has a number of categories associated with it, which can be considered as anchor points within the ontological space. For example, `d:Bar_(music)` is associated with the categories `c:Musical_notation` and `c:Rhythm`.

We consider the following definitions:

- T is the set of all terms with t being a specific term, e.g. `bar`;
- I is the set of all interpretations with i being a specific interpretation, e.g. `d:Bar_(music)`;
- C is the set of all categories with c being a specific category, e.g. `c:Rhythm`;
- $I(t) \in \wp(I)$ is the term to interpretations assignment, where $\wp(I)$ is the powerset of all interpretations, e.g. $I(\texttt{bar}) = \{\texttt{d:Bar\_(music)}, \texttt{d:Bar\_(unit)}\}$;
- $g(i)$ is the interpretation weight;
- $C(i) \in \wp(C)$ is the interpretation to categories assignment, where $\wp(C)$ is the powerset of all categories, e.g. $C(\texttt{d:Bar\_(music)}) = \{\texttt{c:Musical\_notation}, \texttt{c:Rhythm}\}$.

We assume that we have a vector space in which we can assign to each category c a vector $\vec{c}$. We then define an interpretation vector $\vec{i}$:

$$\vec{i} = \frac{g(i)}{\| \sum_{c \in C(i)} \vec{c} \|} \sum_{c \in C(i)} \vec{c}$$

We consider the similarity between two interpretations as being the cosine similarity between the two associated vectors. We consider the similarity between two documents as being the cosine similarity between the weighted sum of interpretation vectors in each document. We define how we construct those weights and the vector space for our interpretation vectors in the following section.

4. AUTOMATED TAGGING OF SPEECH AUDIO

In the following, we propose a method to use the audio in order to automatically assign tags to programmes within the archive, with those tags being drawn from the Linked Data cloud. We start by transcribing the audio and identify terms within the transcripts that could correspond to potential tags. We then build an eTVSM-based model enabling us to disambiguate those terms and rank the corresponding tags. A depiction of the workflow of our automated tagging system is available in Figure 1.

4.1 Automatic transcription

After investigating the various open-source options for multiple speaker automated speech recognition [16] in the context of broadcast news, we settled on the open source CMU Sphinx-3 software, with the HUB4 acoustic model [24] and a language model extracted from the GigaWord corpus[4]. The full set of parameters used by our system is available in Section 8, and was chosen for both speech recognition accuracy (minimal word error rate, or WER) and processing speed (how much faster than real time the transcribing process is).

The results of this speech recognition step are very noisy. The WER in those transcripts varies a lot from programme to programme, depending on the year the programme was recorded in, the accent of the different speakers in the programme, and the background noise in the programme. An average value of the WER for two programmes respectively from 1981 and 2011 is of 47%. However, the WER can go up to 90% on radio dramas that have lots of background noise and different speakers. A full study of the WER obtained on the World Service archive remains to be done.

The WER reported by the THISL system and their ABBOT speech recognition component [1] is of 36.6%. The difference in WER is due to two factors. Firstly, the dataset on which the THISL system works does not span several decades, hence there is less disparity in terms of accents and topics. The THISL dataset is also only holding news programmes, which makes it less heterogeneous in terms of programme genres than the World Service archive. Secondly, a

[3] We use the namespaces defined in Section 9.

[4] See `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05`, last accessed November 2011

specific acoustic model and language model was trained for this particular dataset within THISL, i.e. news outputs from 1998 and 1999. We use an off-the-shelf recogniser (CMU Sphinx-3), acoustic model (HUB4) and language model (GigaWord).

In the following, we try to mitigate the noisiness of those transcriptions in order to derive an accurate list of tags to be applied to the programme.

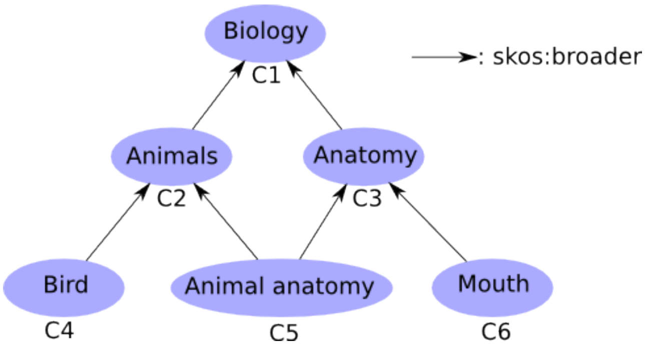

Figure 2: An example SKOS category hierarchy

4.2 Definition of our eTVSM

We now define our Enhanced Topic-based Vector Space Model, which we use for disambiguating and ranking candidate terms extracted from the transcripts.

4.2.1 Terms and interpretations

We start by generating a list of web identifiers used by BBC editors to tag programmes. Those web identifiers identify people, places, subjects and organisations within DBpedia. For each of those identifiers, we dereference them and get their label from their `rdfs:label` property. We strip out any disambiguation string from the label, and apply the Porter Stemmer algorithm [20] to it in order to get to a corresponding term. This defines our set of terms T.

We consider the set of all DBpedia web identifiers as our set of interpretations I.

We store, for each stemmed label, the set of web identifiers it could correspond to. This gives us our term to interpretations assignment $I(t)$.

We define the interpretation weight as follows:

$$g(i) = \frac{1}{|j : j \in I(t), t \in T(i)|}$$

where $T(i)$ is the set of terms corresponding to an interpretation i. The weight of an interpretation will be inverse proportional to the number of possible interpretations of the corresponding terms.

4.2.2 Categories

For a given interpretation i, we construct $C(i)$ using the following SPARQL query:

```
SELECT ?category WHERE { {
  <i> <http://purl.org/dc/terms/subject> ?category
 } UNION {
  <i> _:p _:o .
  _:o <http://purl.org/dc/terms/subject> ?category
} }
```

We include the categories of neighbouring resources to increase possible overlap with other resources mentioned in the same programme. Our evaluation in Section 5 shows that such an expansion gives the best results.

4.2.3 Vector space model for SKOS categories

We now consider the subject classification in DBpedia derived from Wikipedia categories and encoded as a SKOS model. We create a vector space in which all items in that categorisation system will have a representation, which defines our eTVSM.

There are many options for constructing such a vector space. We focus on the one that gave the best results, and provide some evaluation results for a few alternatives in Section 5.

We consider the hierarchy induced by the `skos:broader` property in the DBpedia SKOS model. The set of all items in that hierarchy is our set of categories C. We consider a vector space where each dimension $(c_1, ..., c_n)$ corresponds to one of the n elements of C.

For each category $c \in C$, we consider the set of its ancestors $P(c,k) \in \wp(C)$ at a level k. We then construct a vector $\vec{c}$ as follows:

$$\vec{t} = (\sum_{k=0}^{\beta} \sum_{c_1 \in P(c,k)} \alpha^k, ..., \sum_{k=0}^{\beta} \sum_{c_n \in P(c,k)} \alpha^k), \vec{c} = \frac{1}{\|\vec{t}\|}\vec{t}$$

Each category vector will be non null on the dimensions corresponding to its ancestors. Two categories that do not share any ancestor will have a null cosine similarity. The further away a common ancestor between two categories is, the lower the cosine similarity between those two categories will be. The constant α is an exponential decay, which can be used to influence how much importance we attach to ancestors that are high in the category hierarchy. The constant β can be used to limit the level of ancestors we want to consider. Very generic categories won't be very useful at describing a possible interpretation and discriminating between them.

For example, if we consider the SKOS hierarchy depicted in Figure 2, a value of α of 0.5 and a value of β set to more than 2, we get the vectors in Table 1. We give a few of their pairwise cosine similarities in Table 2.

We now have a vector space in which we can assign each category c to a vector $\vec{c}$. An Open Source implementation of such a vector space applicable to any hierarchy encoded as RDF is available online[5].

4.3 Using the eTVSM for automated tagging

Now our eTVSM model is defined, we use it for identifying potentially relevant terms, disambiguating them and ranking them, in order to identify the most relevant tags to apply to each programme

We start by looking for terms belonging to T in the automated transcripts, after applying the same Porter Stemmer algorithm to them. The output of this process is a list of candidate terms with time-stamps and a list of possible interpretations for those terms, captured as a list of DBpedia web identifiers.

For each programme p in our corpus P, we derive a 'main topic' vector $\vec{t_p}$ from all possible interpretations of all terms:

[5] See `https://github.com/bbcrd/rdfsim`.

	C1	C2	C3	C4	C5	C6
$\vec{C}1$	1	0	0	0	0	0
$\vec{C}2$	$\frac{0.5}{\sqrt{1.25}}$	$\frac{1}{\sqrt{1.25}}$	0	0	0	0
$\vec{C}3$	$\frac{0.5}{\sqrt{1.25}}$	0	$\frac{1}{\sqrt{1.25}}$	0	0	0
$\vec{C}4$	$\frac{0.25}{\sqrt{1.3125}}$	$\frac{0.5}{\sqrt{1.3125}}$	0	$\frac{1}{\sqrt{1.3125}}$	0	0
$\vec{C}5$	$\frac{0.5}{\sqrt{1.75}}$	$\frac{0.5}{\sqrt{1.75}}$	$\frac{0.5}{\sqrt{1.75}}$	0	$\frac{1}{\sqrt{1.75}}$	0
$\vec{C}6$	$\frac{0.25}{\sqrt{1.3125}}$	0	$\frac{0.5}{\sqrt{1.3125}}$	0	0	$\frac{1}{\sqrt{1.3125}}$

Table 1: Values of the different category vectors in the example SKOS hierarchy. The values on the diagonal are all equal to 1 before normalisation.

$\cos(C4, C5)$	0.247
$\cos(C4, C6)$	0.048
$\cos(C5, C6)$	0.247

Table 2: Approximate pairwise cosine similarities between category vectors in the example SKOS hierarchy

$$\vec{t_p} = \sum_{i \in I} w_{p,i} \vec{i}$$

$w_{p,i}$ is the weight assigned to the interpretation i in the programme p. We set it to the term frequency of the terms associated with i in the automated transcript of that programme.

Wrong interpretations of specific terms will account for very little in the resulting vector, while web identifiers related with the main topics of the programme will overlap and add up.

We use this vector for disambiguation. For a given term t, we choose the interpretation $i \in I(t)$ which maximises the cosine similarity between $\vec{i}$ and $\vec{t_p}$.

Then, we use the following $r_{p,i}$ value to rank the different interpretations i according to how relevant they are to describe a particular programme p:

$$r_{p,i} = w_{p,i} * log(\frac{|P|}{|p : t \in p|}) * \frac{\vec{t_p}\vec{i}}{\|\vec{t_p}\|\|\vec{i}\|}$$

This corresponds to the TF-IDF score, weighted by the cosine similarity of the chosen interpretation to the main topic vector.

We end up with a ranked list of DBpedia web identifiers, for each programme. Some examples of the top three tags and their associated scores are given in Table 3, for different programmes.

5. EVALUATION

In this section, we evaluate the above algorithm for automated tagging of speech audio.

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 3: 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.

5.1 Evaluation dataset

We want to compare our automatically extracted tags with tags applied by professional editors. Such tags are made available through the bbc.co.uk/programmes API [22]. We apply the following automated interlinking heuristics to find equivalences between programmes within bbc.co.uk/programmes and the World Service radio archive. If two programmes share the same brand name (e.g. 'From Our Own Correspondent') and the same broadcast date (e.g. 2011-05-11), we assume they are identical.

As a mapping heuristics for programmes data, it works more accurately than matching on episode titles, as they differ a lot from one database to the other. Brand names will usually be the same across databases. We restrict the mapping to programmes that have tags within bbc.co.uk/programmes.

This results in a set of 132 equivalences between programmes in the World Service radio archive and editorially tagged programmes within bbc.co.uk/programmes.

In that dataset, the average number of editorial tags by programme is 4.92, and 477 distinct tags are used. The editorially applied tags are generally of good quality, covering all topics a programme covers. A distribution of the editorially applied tags is available in Figure 3. This distribution exhibits a very long tail, as 377 tags are used only once.

5.2 Evaluation metric

We want our evaluation metric to capture how likely it is for our algorithm to output as its first N tags the N tags applied manually by editors. We use the TopN measure introduced by Berenzweig et al. in [3]:

$$\texttt{TopN} = \frac{\sum_{\texttt{j=1}}^{\texttt{N}} \alpha_{\texttt{r}}^{\texttt{j}} \alpha_{\texttt{c}}^{\texttt{k}_{\texttt{j}}}}{\sum_{\texttt{i=1}}^{\texttt{N}} \alpha_{\texttt{r}}^{\texttt{i}} \alpha_{\texttt{c}}^{\texttt{i}}}$$

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_r and α_c are two exponential decay constants, expressing how much we want to penalise a tag for appearing

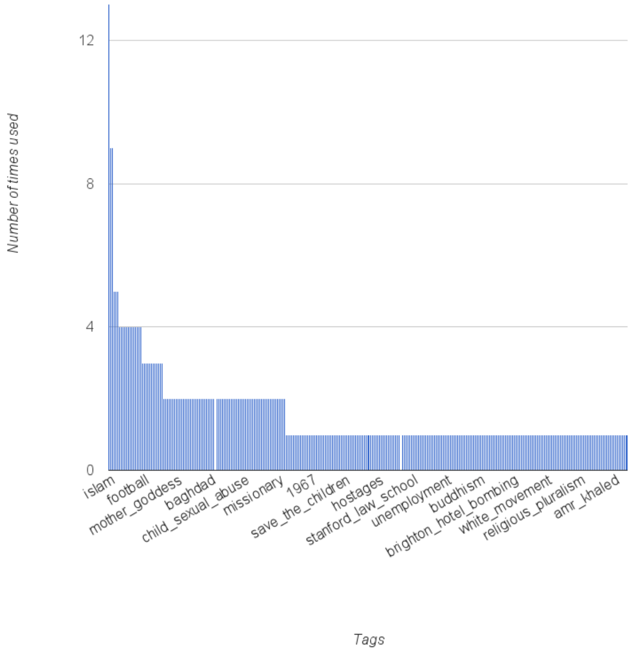

Figure 3: Editorial tags distribution in the evaluation dataset

down the lists.

As the list of tags applied by editors is not ordered, we simplify this measure as follows:

$$\texttt{TopN} = \frac{\sum_{\texttt{j=1}}^{\texttt{N}} \alpha_c^{\texttt{k}_\texttt{j}}}{\sum_{\texttt{i=1}}^{\texttt{N}} \alpha_c^{\texttt{i}}}$$

A score of 1 will be achieved if the tags applied by editors are the top tags extracted by our algorithm. A score of 0 will be achieved if none of the tags applied by editors appear in the tags extracted by our algorithm. We choose a value of 0.8 for our constant α_c, which means that a tag will contribute around 0.1 to the overall score before normalisation if it appears at the tenth position.

We choose this evaluation metric as it captures best the intent of our algorithm. We want editors to skim through the list of automatically extracted tags, add and/or delete from them, and approve them. Therefore, we want the tags most likely to be approved at the top of the list of automatically extracted tags. Precision and recall would not appropriately capture that intended use.

5.3 Evaluation results and discussion

On our evaluation dataset, we get the average `TopN` scores in Figure 4. We got our best result (`TopN = 0.209`) for $\alpha = 0.9$ and $\beta = 10$. We show in Table 4 an example of a good and a bad result, with their associated `TopN` scores.

In Table 5, we also give results for a few variations of our algorithm, for the values of α and β that maximise the score of our tagger when they apply:

- No SKOS expansion — When not expanding the categories associated to a DBpedia web identifier by following forward links, the results obtained were slightly lower;

Editorial tags	Automatic tags
Programme 1, `TopN = 0.242`	
`d:Crime_fiction`	`d:DNA`
`d:DNA`	`d:Double_helix`
`d:Double_helix`	`d:Francis_Crick`
Programme 2, `TopN = 0`	
`d:BP`	`d:Methane`
`d:Climate_change`	`d:Water`
`d:Greenhouse_gas`	`d:Natural_gas`

Table 4: Examples of editorial tags and top automatically applied tags, for two programmes, along with corresponding `TopN` measure for each programme.

- Double SKOS expansion — The best results we had were obtained by expanding the SKOS categories associated with a DBpedia web identifier using both forward and backward links. However, the average number of categories per DBpedia web identifier made the algorithm run very slowly. We decided to compromise on the quality of the results to get our algorithm working in a reasonable time;
- Principal Component Analysis (PCA) — We construct a vector space where each dimension corresponds to a category in the DBpedia SKOS hierarchy, and where each DBpedia web identifier has a corresponding vector, capturing the adjacency of that web identifier to SKOS categories. We use PCA to reduce the dimensionality of that space, and derive similarities between interpretations from cosine similarities in that reduced space. This version of the algorithm scored lower than the approach described above, but had the advantage of being faster, as the resulting space was of much lower dimensionality.

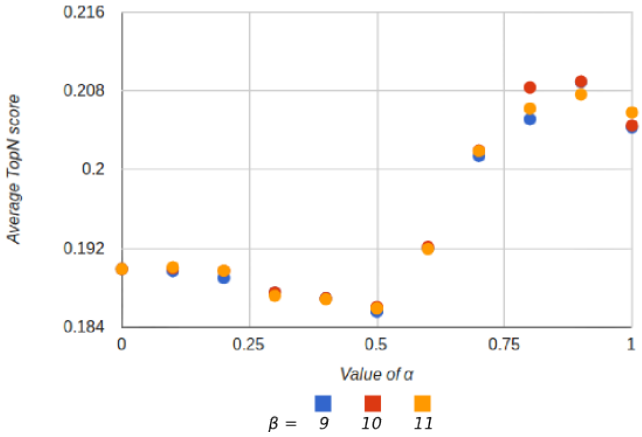

Figure 4: Average `TopN` scores for our automated tagger on our evaluation dataset, for different values of α and $\beta \in \{9, 10, 11\}$. We ommit other values of β giving lower scores for readability.

No SKOS expansion	0.1758
Double SKOS expansion	0.2128
PCA (20 principal components)	0.1351

Table 5: `TopN` measure for different approaches for our automated tagging algorithm

Baseline random tagger	0.0002
DBpedia Spotlight	0.0293
Alchemy API	0.1951

Table 6: `TopN` measure for third-party services

In Table 6, we apply the same evaluation to a baseline tagger picking tags at random and two third-party services.

The first one is DBpedia Spotlight [13]. We submitted the output of the transcription for two minutes chunks of programmes to the DBpedia Spotlight API. We then summed the scores of the entities returned by the API across the length of the programme. Finally, we applied the same inverse document frequency step as in our algorithm, in order to normalise the DBpedia Spotlight results across the entire corpus. It appears that DBpedia Spotlight does not work well with noisy text, outputted by an automated transcription process. In particular, the disambiguation process in DBpedia Spotlight relies on the text surrounding a particular term to be relatively clean. The transcripts being very noisy, that process mostly returns the wrong interpretations. It also appears that DBpedia Spotlight relies heavily on capitalisation, however capitalisation is not available in the automated transcripts. It is also important to note that DBpedia Spotlight tackles a different problem. It extracts entities from text but does not try to describe an entire document using a few selected entities.

It appears that using the structure of DBpedia itself for disambiguation gives satisfying results: deriving a model of a main programme topic from all possible interpretations of all relevant terms, and picking the interpretations that are closest to that main topic. Mis-interpretations will account to very little in that main topic vector, as most of them will be very dissimilar to each other.

We tried a number of commercial third-party concept tagging APIs, and the result of the one that scored the best in our evaluation is also shown in Table 6. We applied the same methodology as for DBpedia Spotlight, so that this third-party service can also benefit from information about the whole corpus. This third-party service performs almost as well as our algorithm. However, no information is publicly available on how that service works.

6. CONCLUSION AND FUTURE WORK

In this paper, we described an automated interlinking system for radio programmes with web identifiers from the Linked Data cloud. We use an Enhanced Topic Vector Space Model to disambiguate and rank candidate terms, identified within automated transcripts. We evaluated this system against tags manually applied by editors. The results, although by no means perfect, are good enough to efficiently bootstrap a tagging process. As the resulting tags are Linked Data web identifiers, isolated archives can effectively be interlinked with other datasets in the Linked Data Web.

We describe in [21] the process of applying this framework to the entire World Service archive, and an application using automatically extracted tags to aid discovery of archive content.

Future work includes creating an editorial interface to enable editors and the public to edit and approve the list of automatically derived tags. We also want to try and incorporate more data (synopsis, broadcast dates, etc.) in the automated tagging process when this data is available. We could also enhance our results by considering more textual representations for DBpedia identifiers than their labels, using similar methodologies as in [13]. We also want to improve the results of the automated speech recognition step, by creating an acoustic model for British English, and a language model built from programme transcripts. We have access to a large pronunciation database maintained within the BBC, and holding about 50 years worth of topical entities with associated BBC pronunciation, which might be useful to construct a better pronunciation dictionary. We also want to study the impact of the noisiness of the transcripts on the results of our algorithm.

The tagging process outputs tags with a time-stamp. We are currently investigating using these time-stamped tags as a basis for topic segmentation. Each tag has a position in the vector space constructed above, and we can track how the average position in that space evolves over time, giving an idea as to when the programme changes topics. Such a segmentation could also be used to feed back in the topic model described in this paper — the topics will be more consistent in each of these segments.

Rather than relying on a SKOS hierarchy, it would also be interesting to find a more broadly applicable way of projecting Linked Data in a vector space, based on the adjacency matrix of the Linked Data graph considered. The PCA-based approach mentioned in the evaluation section would be a good starting point, but would need to be made more robust.

7. 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.

8. ANNEX: SPHINX-3 PARAMETERS

```
-samprate   28000
-nfft       1024
-beam       1e-60
-wbeam      1e-40
-ci_pbeam   1e-8
-subvqbeam  1e-2
-maxhmmpf   2000
-maxcdsenpf 1000
-maxwpf     8
-ds         2
```

9. ANNEX: NAMESPACES

```
@prefix d: <http://dbpedia.org/resource/> .
@prefix c: <http://dbpedia.org/resource/Category:> .
```

10. 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] Soren Auer, Christian Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the International Semantic Web Conference*, Busan, Korea, November 11-15 2007.

[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] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3(3):993–1022, March 2003.

[5] 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.

[6] Sam Davies, Penelope Allen, Mark Mann, and Trevor Cox. Musical moods: A mass participation experiment for affective classification of music. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011.

[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] 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.

[9] D. Kuropka. *Modelle zur Repräsentation natürlichsprachlicher Dokumente - Information-Filtering und -Retrieval mit relationalen Datenbanken*. Logos Verlag, 2004. ISBN: 3-8325-0514-8.

[10] John Makhoul, Francis Kubala, Timothy Leek, Daben Liu, Long Nguyen, Richard Schwartz, and Amit Srivastava. Speech and language technologies for audio indexing and retrieval. *Proceedings of the IEEE*, 88(8):1338–1353, August 2000.

[11] Korn El Mark, Kornél Markó, Udo Hahn, Stefan Schulz, and Percy Nohama. Interlingual indexing across different languages. In *RIAO 2004 – Conference Proceedings: Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, pages 82–99, 2004.

[12] Olena Medelyan, Ian H. Witten, and David Milne. Topic indexing with wikipedia. *Proc. of Wikipedia and AI workshop*, 2008.

[13] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.

[14] Alistair Miles, B. Matthews, M. Wilson, and D. Brickley. SKOS core: Simple knowledge organisation for the web. In *Proceedings of the International Conference on Dublin Core and Metadata Applications (DC-2005),*, pages 5–13, Madrid, 2005.

[15] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press. paper presented at the poster track.

[16] Partrick Nguyen. Techware: Speech recognition software and resources on the web. *IEEE Signal Processing Magazine*, pages 102–108, 2009.

[17] J. Scott Olsson and Douglas W. Oard. Improving text classification for oral history archives with temporal domain knowledge. In *SIGIR'07*, 2007.

[18] G. Paaß, E. Leopold, M. Larson, J. Kindermann, and S. Eickeler. SVM classification using sequences of phonemes and syllabes. In *Principles of Data Mining and Knowledge Discovery*, pages 373–384, 2002.

[19] Artem Polyvyanyy. Evaluation of a novel information retrieval model: eTVSM. Master's thesis, Hasso Plattner Institut, 2007.

[20] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[21] Yves Raimond, Chris Lowis, and Jonathan Tweed. Automated semantic tagging of speech audio. In *Proceedings of the WWW'12 Demo Track*, 2012.

[22] 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, page 291. Springer, 1st edition edition, 2010.

[23] Richard C. Rose, Eric I. Chang, and Richard P. Lippmann. Techniques for information and retrieval from voice messages. In *ICASSP'91*, pages 317–320, 1991.

[24] 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.

[25] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago - a core of semantic knowledge. In *16th international World Wide Web conference*, 2007.

[26] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. Kea: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, pages 254–255, New York, NY, USA, 1999. ACM.