Quote Extraction and Analysis for News

Chris Newell
BBC Research & Development
London, UK
chris.newell@bbc.co.uk

Tim Cowlishaw
BBC Research & Development
London, UK
tim.cowlishaw@bbc.co.uk

David Man
BBC Research & Development
London, UK
david.man@bbc.co.uk

ABSTRACT

Quotes are a key element of journalism, where news stories are frequently substantiated by quoting a number of named sources. This paper describes a system which automatically extracts quotes from news feeds and archives to provide a structured database of claims and opinions. Context metadata is extracted from the news articles to allow the extracted quotes to be easily searchable for purposes such as journalistic research. Statistical analysis of the database can provide insights into how particular quote sources and topics are used. We also illustrate how the database could be used for the comparative analysis of quotes, potentially allowing contradicting claims or shifting opinions to be identified.

CCS CONCEPTS

• **Information systems → Information extraction**; *Data mining*;

KEYWORDS

ACM proceedings, computational journalism

ACM Reference Format:
Chris Newell, Tim Cowlishaw, and David Man. 2018. Quote Extraction and Analysis for News. In *Proceedings of KDD Workshop on Data Science, Journalism and Media (DSJM)*. ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

News organisations have experienced rapid changes in recent years as new means of distribution have arisen, leading to changing patterns of consumption. Journalists have also faced significant challenges from the rising breadth and complexity of information sources, whilst facing the need to maintain broad coverage of a range of news topics with limited resources. Meanwhile, concerns have arisen about the spread of false or misleading information in the traditional and less well-established media, creating the need to identify the sources of claims and assertions and to understand how these have developed over time.

Recent advances in Machine Learning and Natural Language Processing provide an opportunity to support journalists faced with these challenges. This paper explores one such opportunity, namely the automatic extraction, categorisation and analysis of quotes from news feeds and archives.

Quotes are a key element of journalism, where news articles are frequently substantiated by quoting a number of named sources. The ability to search and identify quotes found in news feeds and archives is therefore of great interest to journalists creating news content. However, quote extraction and analysis is also of interest to those wanting to verify the accuracy of news, by checking the consistency and reliability of quoted sources.

In the first section of this paper we describe the automated extraction process and supporting functions such as coreference resolution. In the second section we explore how these functions can be introduced into an integrated system, addressing user requirements developed with journalists. Finally, we illustrate how the database can be used to enable the comparative analysis of quotes and discuss potential applications.

DSJM, August 2018, London, UK

2 RELATED WORK

Most of the earlier work on quote extraction [1–3] has focused on direct quotes, which are delimited by quotation marks. However, these typically represent only 30% of the quotes found in news articles. More recent work by Pareti et al [4] has also addressed indirect quotes (without quotes) and mixed quotes (with direct and indirect parts). We adopt a similar approach in this paper with some enhancements.

For training and testing we use the Penn Attribution Relations Corpus (PARC) [5] which is an extension to the attribution annotations included in the Penn Discourse Treebank [6]. The corpus covers around 2.2K Wall Street Journal articles and provides over 16K annotated quotes.

3 QUOTE EXTRACTION

A quote typically consists of three elements: the source, a verb-cue and one or more content spans. We identity each of these elements individually using a dedicated classifier.

3.1 Verb-cue classifier

The verb-cue in a quote is usually a speech verb (e.g. "said") but they can vary widely. For this reason, we follow the approach proposed in [4] and use a classifier rather than a look-up table to identify verb-cues. However, we use a Convolutional Neural Network (CNN) classifier provided by the spaCy NLP library [7] rather than the k-NN classifier used in [4] as it provides better performance. The CNN classifier is designed for Named Entity Recognition but is an effective verb-cue classifier when trained with PARC annotations.

The performance of the CNN verb-cue classifier is shown in Table 1. The predicted span is considered to be correct if it is an exact match with the annotations in PARC. This metric is used for all the results presented in this paper. For comparison, the k-NN classifier described in [4] achieved an F1 Score of 79.9%.

The performance of the verb-cue classifier is critical to the overall performance of the system, as all the other classifiers required for

Table 1: Performance of the verb-cue classifier

	Precision	Recall	F1
Verb-cue Classifier	97.9%	85.8%	91.5%

the extraction and attribution process use the location of verb-cues as a feature.

3.2 Quote content and source classifiers

To identify quote content spans we use the Token-based approach described in [4] and predict labels using CRFSuite [8], a fast implementation of Conditional Random Fields. We also use the spaCy NLP library to provide tokenization, Part of Speech tagging, dependency parsing and Named Entity Recognition. The quote content classifier is trained using the following features for each token:

- the text of the token
- the lemma of the token
- the spaCy POS tag
- the spaCy TAG tag
- the spaCy entity IOB label
- the text of the previous five tokens
- the text of the next five tokens
- whether the token is in quotation marks
- the dependency depth
- the dependency relation
- whether the token is the child of a verb-cue
- whether the token is the leftmost child of a verb-cue
- whether the token follows a verb-cue
- whether the sentence has a verb-cue
- the index of the token in the sentence

Each token in the training data is given either a I, O or B label, where a B tag identifies the first token in a content span, I tags identify the remaining tokens in the span and O labels identify tokens outside content spans.

Whereas previous work [1–4] has assumed that candidate quote sources are simply the entity mentions found within the document, we use a dedicated quote source classifier. This is again implemented using CRFSuite and trained with IOB labels derived from PARC. We use the same features as used for the quote content classifier with the following additions:

- the label predicted by the content classifier
- the spaCy entity type
- the distance from the verb-cue (if a dependent of one)
- whether the token is the rightmost child of a verb-cue

We found this approach usefully restricted the number of candidates and captured more detailed characteristics of the sources. For example, the source classifier might identify a source as "a spokesperson for HMRC" whereas the entity mention would simply be “HMRC”. The performance of the quote content and source and classifiers is shown in Table 2.

Table 2: Performance of the content and source classifiers

	Precision	Recall	F1
Content Classifier	76.4%	68.6%	72.3%
Source Classifier	92.6%	89.8%	91.2%

Table 3: Example output from the verb-cue classifier (VCC), quote source classifier (QSC) and quote content classifier (QCC) using IOB labelling

TEXT	VCC	QSC	QCC
The	O	B	O
Russian	O	I	O
government	O	I	O
announced	B	O	O
in	O	O	O
January	O	O	O
that	O	O	B
VPN	O	O	I
providers	O	O	I
would	O	O	I
need	O	O	I
to	O	O	I
obtain	O	O	I
a	O	O	I
licence	O	O	I
to	O	O	I
distribute	O	O	I
their	O	O	I
products	O	O	I
.	O	O	O

The operation of the three classifiers is illustrated in Table 3 which shows an example of an indirect quote. In general, the classifiers are reliable with straightforward sentences of this kind. However, more complex grammatical structures are more likely to lead to errors.

4 QUOTE ATTRIBUTION

Having identified the individual elements of the quotes within a document, the relationship between the elements needs to be established by attributing the quote content to specific sources.

4.1 Content and source resolvers

For quote attribution we use a similar approach to the No Seq. method described by Tim O’Keefe et al [9]. However, we adopt a two-stage process where we first identify the verb-cue associated with each content span using a classifier called the content resolver. We then identify the source span associated with each verb-cue using a classifier called the source resolver. This two-stage approach supports cases where a quote has multiple content spans but also cases where a single source is associated with multiple verb-cues (e.g. He said “...” adding that “...”).

Table 4: Performance of the content and source resolvers

	Precision	Recall	F1
Content Resolver	99.5%	93.6%	96.5%
Source Resolver	97.5%	91.8%	94.6%

Table 5: Performance of the coreference resolver

	Precision	Recall	F1
Personal Pronouns	89.8%	86.8%	88.3%

Both the content resolver and the source resolver use Max Entropy classifiers trained using PARC annotations. The content resolver uses the following text features:

- the distance (in words) between the content span and the verb-cue
- whether the content span and the verb-cue are in the same sentence
- whether the content span is a descendent of the verb-cue

The source resolver uses the following text features:

- the distance (in words) between the source span and the verb-cue
- whether the source span and the verb-cue are in the same sentence
- whether the source span and verb-cue are in the same parenthetical phrase (delineated by commas)

The performance of the two resolvers is shown in Table 4.

4.2 Coreference resolution

The final step in the quote attribution process is to resolve any abbreviated names or personal pronouns found in the quote sources which are coreferent with entity mentions found earlier in the document. Initially, we tried standard coreference libraries to resolve these coreferences but found these to be slow, since they attempt to resolve all potential coreferences within a document. Instead, we developed a dedicated coreference resolver.

We first cluster all names found in the document, including abbreviated forms, by applying a simple rules-based approach which uses surname and/or forename matching. We also resolve any personal pronouns found in the quote sources using the approach proposed by Greg Durrett et al [10].

For training data we use the intersection of PARC and the CoNLL-2011 (Conference on Computational Natural Language Learning) Shared Task dataset [11] which includes coreference annotations. We use this to train a Max Entropy classifier, using the text features proposed in [10]. This classifier identifies the most likely entity to be associated with each personal pronoun.

The performance of the coreference resolver for personal pronouns is shown in Table 5, measured using the intersection of PARC and CoNLL-2011 test data.

Table 6: Overall performance of the quote extraction system

	Precision	Recall	F1
Overall Performance	62.1%	52.2%	56.8%

4.3 Overall performance

The overall performance of the quote extraction system is shown in Table 6. This is measured using the intersection of PARC and CoNLL-2011, requiring an exact match for all the quote spans after resolving coreferences. This is a stringent metric and in many cases there may a reasonably accurate, but partial match of the spans.

Both the quote attribution and coreference resolution processes have a significant error rate but we believe this to be sufficiently low enough for our application, where the results can be checked and corrected by the user. However, this will need to be confirmed in operational use. The speed of the combined extraction and attribution processes is sufficient to allow large news archives (e.g. 260K articles) to be processed in one or two days.

5 USER REQUIREMENTS

After establishing that automated quote extraction and attribution is technically feasible we conducted a series of interviews with journalists to understand their workflow and to establish how providing access to quotes might be useful for their work.

Initial feedback established that simply highlighting the quotes in individual documents would be a useful and time-saving feature. This function would also be important to allow the accuracy of extracted quotes to be checked in the context of the original document.

Journalists are often looking for the best 'news-making quotes'. A natural requirement of this is the need to search for quotes concerning specific topics or mentioning specific entities. This led us to consider the context information we needed to acquire and associate with individual quotes.

Additional requirements focused on topics covered by individual sources. Journalists were interested in seeing what a particular person has been talking about and more interestingly how that person's quotes about a given topic have changed over time. This feature was seen as being very useful when writing features or analysis pieces.

Journalists also wanted to be able to compare quotes on a given topic from different publications, particularly where it was possible to identify clear differences of fact or opinion. Finally, the ability to extract quotes from non-news feeds like reports and press releases was also desired.

When a completely new technical possibility arises it is important to get feedback from the potential end users. However, they may not initially appreciate how they would use the technology without first having some experience of it. In our system development work it was important to demonstrate the editorial capabilities as simply and clearly as possible. Our approach has been to build a prototype incorporating our initial ideas and user requirements. We then add features and update the design as we get feedback. This is an iterative process where we learn and adapt as we better understand the needs and expectations of the end users.

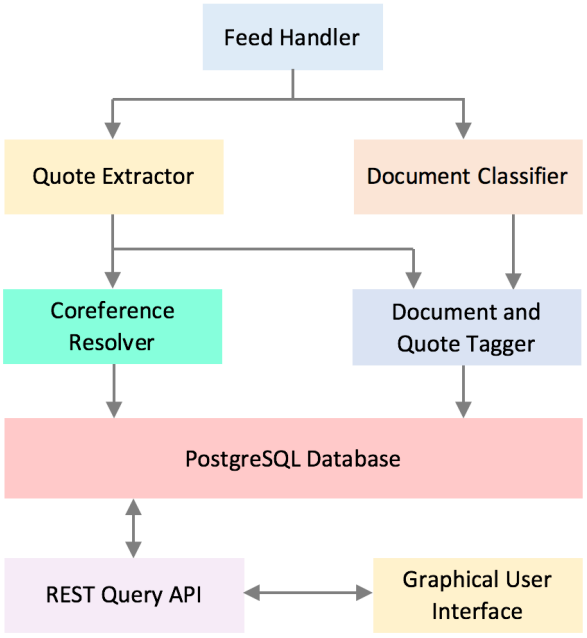

Figure 1: Functional diagram of the quote database.

6 SYSTEM DESIGN

The design of our integrated prototype is shown in Figure 1. In order to provide as much context information as possible for each quote we support two processing pipelines: one on the left, for quote extraction and attribution; the other on the right, for metadata extraction. Text from the quote extractor pipeline is passed to the tagger in the metadata extractor pipeline.

The metadata pipeline includes a news document classifier and a semantic tagging system. The document classifier is intended to identify the news topics covered by the articles from which the quotes are extracted. The semantic tagger is intended to identify entity mentions within each individual quote.

The news document classifier is a multi-label classification system, trained using a corpus of around 500K news and sport articles which have been manually classified by BBC journalists. The classifier supports over 10K unique topics from the BBC Things ontology [12] using a One-versus-Rest approach [13] with individual SVM classifiers for each class. The classifier typically identifies 3-5 topics for each news article and we make the assumption that these topics are relevant to all the quotes extracted from an article.

The semantic tagger is a dictionary based system, similar to DBpedia Spotlight [14] which can identify mentions of people, organisations and places etc. For each candidate tag a confidence score is calculated using a vector word model which is used to compare the tag vector, firstly with the mention text and secondly with the document vector. The confidence score is used to reduce the number of spurious entities and to resolve ambiguous cases. We apply the tagger to the complete document, since this gives the most accurate disambiguation results, and then identify tags which occur within individual quote content spans. The tags are natively DBpedia identifiers but we map these to BBC Things.

Table 7: Characteristics of the 3-year quote database

Element	Count
Articles	260K
Quotes	807K
Sources	25K
Unique Topics	8.4K
Topic Instances	845K
Unique Tags	85K
Tag Instances	1.9M

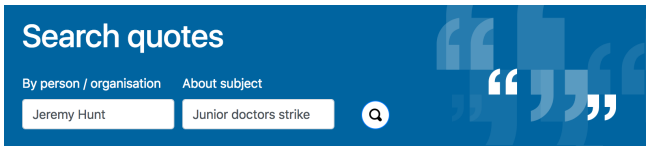

Figure 2: Search interface for the quote database.

The extracted quotes are stored in a PostgreSQL database together with their associated topics and semantic tags. A REST query interface allows external systems to access the database and supports a graphical user interface.

For experimental purposes the system was populated with a 3-year archive of recent BBC News articles. The ingest process was completed in a few days. A summary of the extracted data is shown in Table 7.

7 RESULTS AND OBSERVATIONS

The general purpose search query interface from our prototype is shown in Figure 2 and some example results are shown in Figure 3. Note that all of these results are mixed quotes and that the second result has an unusual verb-cue (“accused”). The system makes it easy to find the earliest or most recent quotes concerning a particular topic and/or by a specific person.

For quote searches involving a specific person, the results include a summary of the topics associated with all their quotes as shown on Figure 4. This allows journalists to see whether the source is a regular contributor on the specified topic or a peripheral contributor whose primary contributions lie elsewhere.

For quote searches involving a specific topic it is possible to determine the frequency of quotes over a period of time. An example of this is shown in Figure 5 where the specified topic is “Brexit”. Similarly, it is possible determine the frequency of quotes from a particular source over time.

To verify the accuracy of the extraction and attribution process an individual quote can be displayed in the context of the original document text, as shown in Figure 6. The source, coreference, cue and content spans are highlighted in different colours.

The primary source of significant errors appears to be the attribution and coreference resolution processes. Attribution errors are frequently found to be associated with more complex grammatical structures such as asides e.g. “Sir Vince Cable, who replaced

Jeremy Hunt 13 Jan 2016	*said*	the number that had gone into work showed "the values of the vast majority of junior doctors" http://www.bbc.com/news/health-35294637
Jeremy Hunt 7 Feb 2016	*accused*	the union of "spreading misinformation" http://www.bbc.com/news/health-35515957
Jeremy Hunt 7 Feb 2016	*said*	the BMA was behaving in a "totally irresponsible way" http://www.bbc.com/news/health-35515957
Jeremy Hunt 27 Feb 2016	*said*	that the national funding formula for GP practices "is clearly adversely affecting the financial stability of single-hander practitioners" http://www.bbc.com/news/uk-england-leicestershire-35671650

Figure 3: Example search results from the quote database.

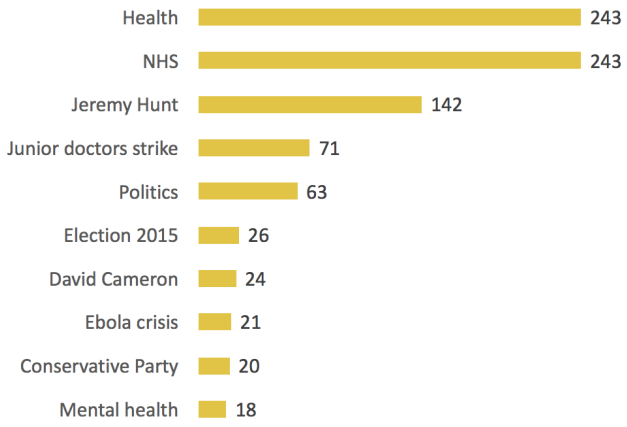

Figure 4: Example of the topics associated with a specific source.

Tim Farron as leader in July, said...". In these cases, the subsidiary nature of the aside may not be recognised by the source resolver. Coreference errors are usually associated with the failure to resolve personal pronouns correctly across multiple sentences.

In some cases quotes are completed missed. This is generally caused by the verb-cue classifier failing to recognise unusual verb-cues. These failures may stem from differences in the vocabulary and expressions used in the PARC training data and our UK-English news archive. The performance of the classifier would probably be improved by broadening the range of training data.

7.1 Sentiment analysis

In the future we hope to use the quote database as a source of data for further analysis with the goal of identifying anomalies

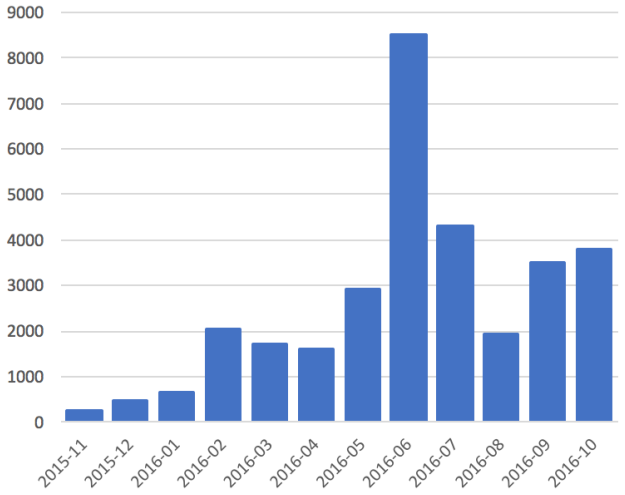

Figure 5: Quotes per month against time where the document topics include "Brexit".

Labour 'staking out new centre ground', says Jeremy Corbyn.
Labour is "staking out the new centre ground in British politics", Jeremy Corbyn has said as the party leaders gave their new year messages. Mr Corbyn said the prospect of a "new Britain" was "closer than ever" and he was leading a "government in waiting". Prime Minister Theresa May used her new year message to say Britons will feel "renewed confidence and pride" in 2018. Lib Dem leader Sir Vince Cable called for a second public

Figure 6: Example quote with individual elements highlighted in the context of the original document.

and contradictions. At this point in time the additional processing has been confined to sentiment analysis of the quote content. For this we have used the sentiment polarity function of the TextBlob library [15] in its default configuration.

Sentiment analysis can help to identify the range of opinions found in a group of sources commenting on a specific topic and can reveal changing trends over time. For example, Figure 7 is a scatter plot of quote sentiment against time for quotes containing a reference to Emmanuel Macron.

It can be seen there was a burst of quotes with a broad range of sentiment around the time of the French Presidential election in April-May 2017. However, it also shows that there has generally been a positive trend in the sentiment during his initial period in office.

The sentiment scores can potentially help users to find individual quotes at different points in the range of opinion. We hope to identify other useful comparative measures.

8 CONCLUSIONS AND FUTURE WORK

In this paper we describe a system which automatically extracts and attributes quotes from news feeds and archives to create a searchable database. We supplement each quote with context information describing both the originating document and the quote content.

The database could either be used directly for journalistic research or as a data source for further analysis by machine learning systems. Statistics derived from the quote metadata can also provide

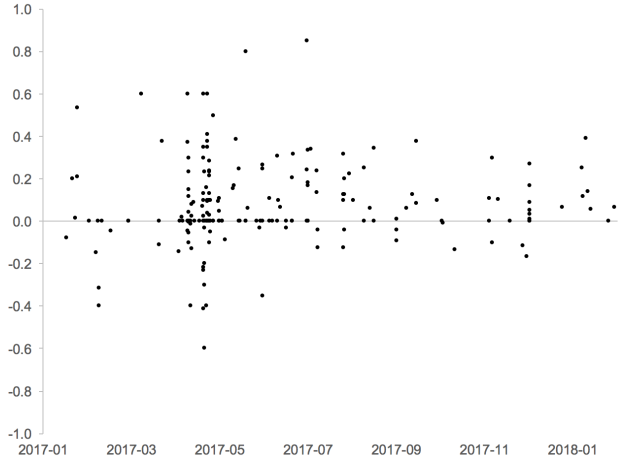

Figure 7: Scatter plot of quote sentiment against time for quotes mentioning Emmanuel Macron.

statistical insights into how quotes and their sources are used by news organisations.

Currently we use a sequence of individual classifiers for quote extraction and attribution. However, we believe a parallel approach would have advantages as it could benefit more from the mutual dependencies between the different elements of a quote.

In future work we hope to explore how the analysis of quote content could be used to help verify the accuracy of news. This could be achieved by checking the consistency and reliability of quoted sources, or by detecting misleading language.

ACKNOWLEDGMENTS

The authors would like to acknowledge the help and advice from Silvia Pareti and the Institute for Language, Cognition and Computation at the University of Edinburgh.

REFERENCES

[1] Ralf Steinberger Bruno Pouliquen and Clive Best. 2007. Automatic detection of quotations in multilingual news. In *Proceedings of Recent Advances in Natural Language Processing*. 487–492.

[2] Kevin Glass and Shaun Bangay. 2007. A naive salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA07)*. 1–6.

[3] David K. Elson and Kathleen R. McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth Conference of the Association for the Advancement of Artificial Intelligence*. 1013–1019.

[4] Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically Detecting and Attributing Indirect Quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 3914–3920.

[5] Silvia Pareti. 2016. PARC 3.0: A Corpus of Attribution Relations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. 3914–3920.

[6] Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie Webber. 2007. *The Penn Discourse Tree-Bank 2.0 annotation manual.* Technical Report. University of Pennsylvania: Institute for Research in Cognitive Science, Cambridge, MA, USA.

[7] Matthew Honnibal. 2018. spaCy EntityRecognizer. Retrieved March 20, 2018 from https://spacy.io/api/entityrecognizer

[8] Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). Retrieved March 20, 2018 from http://www.chokkan.org/software/crfsuite

[9] Tim O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 790–799.

[10] Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1971–1982.

[11] Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task (CoNNL-2011)*. Association for Computational Linguistics, 1–27.

[12] BBC. 2018. BBC Things. Retrieved March 20, 2018 from https://www.bbc.co.uk/things/about

[13] Scikit-learn. 2017. OneVsRestClassifier. Retrieved May 2, 2018 from http://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

[14] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*. 121–124.

[15] Steven Loria. 2017. TextBlob: Simplified Text Processing. Retrieved May 2, 2018 from https://textblob.readthedocs.io/en/dev