Natural Language Engineering **24** (2): 317–324.

doi:10.1017/S1351324918000013

Industry Watch: Text analytics APIs, Part 1: The bigger players

ROBERT DALE
Language Technology Group
e-mail: rdale@language-technology.com

(*Received 3 January 2018*)

Abstract

If you're in the market for an off-the-shelf text analytics API, you have a lot of options. You can choose to go with a major player in the software world, for whom each AI-related service is just another entry in their vast catalogues of tools, or you can go for a smaller provider that focusses on text analytics as their core business. In this first of two related posts, we look at what the most prominent software giants have to offer today.

1 The text analytics landscape

For a recent project, I needed to review text analytics API offerings from a wide range of vendors. These APIs have proliferated over the last few years: the easy availability and elasticity of cloud-based resources means that the barrier to entry for developing a text analytics-driven SaaS product is relatively low, and there are now at least a couple of dozen APIs you can make use of today. So if you're looking to make use of a third-party text analytics functionality rather than build your own, how do you choose which to go with? This post, and the one following it, aim to help if you happen to be faced with that particular quandary.[1] Here, we look at what we might think of as the 'Goliaths' of the software industry: major players for whom text analytics is just one capability amongst a wide range of offerings. You might be drawn to these providers if, for example, you wanted to be confident that the tools you're using are backed by a company that's sure to be around in a few years' time. In a subsequent post, we'll consider the 'Davids': the smaller players who focus specifically on text analytics, and whom you might hope would be more nimble and innovative.

A note on scope: we'll take text analytics to be concerned with the processing of text documents of various kinds and in various ways, typically using functionalities that most in the field would consider to require some form of natural language

[1] All the information provided here was accurate, as far as I can determine, in January 2018; but things can change rapidly, so you'd be best advised to double check any critical information before making a decision based on this post.

processing. The continued high level of interest in chatbots and virtual assistants means that every vendor also has a range of language-related services to support dialogic applications of that type, but that's not what we're considering here.

2 The Goliaths

While there are other software giants who offer text analytics capabilities—for example, HP have their Haven OnDemand services[2] and SAP offer Text Data Processing Services[3]—we focus here on the four major players who are most visible in this space: in alphabetic order, Amazon, Google, IBM and Microsoft. Facebook and Apple also have APIs for some text analytics tasks,[4] but in each case, these are principally intended as tools for working on those companies' respective software platforms. If you're a software developer who wants to build a standalone application that makes use of a third-party text analytics toolset, the four we consider here are probably the most obvious alternatives.

Amazon is the newest member of this club: although they've had various language-related services around the Alexa platform for a while, the text analytics capabilities, packaged as Amazon Comprehend,[5] only became publicly available in November 2017, along with Amazon Translate and Amazon Transcribe. Of course, these are only elements of a massive collection of services provided under the AWS banner. Comprehend offers five distinct text analytics services: entity recognition, sentiment analysis, keyphrase extraction, language detection and topic modelling. Comprehend is categorised as a Machine Learning service, along with the other Alexa-related services.

Google's Natural Language API[6] is part of the Google Cloud platform. The APIs provided cover entity recognition, sentiment analysis, syntactic analysis and content classification. Again, these are only components in a broad suite of cloud-based tools. Google also categorises these functionalities under Machine Learning, along with a range of related tools that include higher level services like 'Job Discovery'.[7] The Google NLP API was released in beta in July 2016.

As we've noted in this column before, IBM's Natural Language capabilities—or at least the ones we're interested in here—come by way of acquisition: IBM bought AlchemyAPI, probably the first cloud-based text analytics API, in March 2015. AlchemyAPI itself was launched in 2009; Alchemy retained its name as an IBM Watson service until early 2016, when it became IBM Natural Language Understanding.[8] The IBM API offers a wider range of capabilities than the other providers discussed here: entity detection, sentiment analysis, topic detection,

[2] https://dev.havenondemand.com/apis/
[3] https://wiki.scn.sap.com/wiki/display/EIM/Text+Data+Processing
[4] See https://developers.facebook.com/docs/messenger-platform/built-in-nlp and https://developer.apple.com/documentation/sirikit respectively.
[5] https://aws.amazon.com/comprehend/
[6] https://cloud.google.com/natural-language/
[7] https://cloud.google.com/products/machine-learning/
[8] https://www.ibm.com/watson/services/natural-language-understanding/

https://doi.org/10.1017/S1351324918000013 Published online by Cambridge University Press

keyword detection, content categorisation in a five-level hierarchy, relation and semantic role extraction, and some useful metadata tools. The IBM framework also lets you extend the provided capabilities with custom domain-specific models for entity and relation identification.

Microsoft's NLP-related offerings are distributed across various subcategories under the Azure platform's Cognitive Services. Under the Language subcategory, we have Linguistic Analysis (which provides tokenisation, POS tagging and constituency-based parsing) and Text Analytics (which provides language detection, sentiment analysis and key phrase extraction), amongst others. The Entity Linking API[9] is found under the Knowledge subcategory, and has been part of the Azure services since at least early 2016; prior to that, it was part of Microsoft's Project Oxford.

Since the specific range of functionalities available varies by provider, your particular needs may rule some candidates out of consideration from the outset.

3 Using the APIs

Because they are all full-stack cloud platform providers, for each of the above you can develop and deploy applications completely in the cloud, without using anything more than a terminal window on your local machine. But you can also use these text analytics services as APIs called from an application hosted on your own machine or elsewhere.

In each case, it's fairly straightforward to sign up for a free or trial account, and to set up authentication to use the services. All four providers offer SDKs and support for various programming languages; again, there's some variation in terms of the languages supported, so that's another factor you might want to consider.

To make comparison simple, we'll focus here on just one specific core capability: detecting named entity mentions in a text document. Each of the four APIs provides slightly different input options for this task; we'll discuss the outputs further below.

- Amazon's approach is the simplest: you provide a text string and the language to be used for analysis. Currently, only English and Spanish are supported.
- Google accepts a plain text or HTML document, or a reference to a document located in Google Cloud storage, and the text encoding type, which is important in terms of calculating offsets. You can also specify the language to be used in analysis; by default this is auto-detected, but texts with lots of non-English names appear to throw this off, so it's safer to specify this parameter if you can.
- IBM's API also accepts plain text or HTML content, or the URL of a document to be analysed, along with the language to be used in the analysis. The API offers a neat feature that attempts to remove advertisements from a retrieved HTML page, and you can ask for the emotion (joy, anger, etc.)

[9] https://azure.microsoft.com/en-au/services/cognitive-services/entity-linking-intelligence-service/

https://doi.org/10.1017/S1351324918000013 Published online by Cambridge University Press

Table 1. *Recognised entity types*

Provider	Entity types
Amazon	CommercialItem, Date, Event, Location, Organization, Other, Person, Quantity, Title
Google	ConsumerGood, Event, Person, Location, Organization, Other, Unknown, WorkOfArt
IBM	Anatomy, Award, Broadcaster, Company, Crime, Drug, EmailAddress, Facility, GeographicFeature, HealthCondition, Hashtag, IPAddress, JobTitle, Location, Movie, MusicGroup, NaturalEvent, Organization, Person, PrintMedia, Quantity, Sport, SportingEvent, TelevisionShow, TwitterHandle, Vehicle

and sentiment (negative or positive) expressed towards the entities that are detected.

- Microsoft's API only supports UTF-8 text. Unlike the others, the Microsoft service expects only a single paragraph of text per call. Also uniquely, the API offers an option whereby you can interrogate a specific word or phrase within a text. These properties make me wonder whether the target use cases the developers had in mind here might be different from the other offerings; for example, a per-paragraph mode of operation might be useful as something you trigger when ENTER is keyed in a word processor.

Perhaps not surprisingly, all these services have slightly different notions of what counts as a named entity. Table 1 lists the types recognised by each. Microsoft's service isn't listed here because it doesn't return type information for the entities it detects; more on that below. Of course, just because two providers use the same name for a category doesn't mean they each define that category in the same way; this leads to difficulties in cross-platform comparison.

The granularity of IBM's categories is striking; but there's more: the service also detects 433 entity subtypes including things like 'BicycleManufacturer' and 'VideoGamePublisher'.[10] A given entity may be assigned several subtypes: for example, as well as being of type 'Company', CNN has the subtypes 'Broadcast, 'AwardWinner', 'RadioNetwork' and 'TVNetwork'.

4 The API outputs compared

The outputs from the four APIs also have their differences, as shown in Figure 1.

Amazon returns an array of entity mentions, each with its begin and end offsets, a type drawn from the list shown in Table 1, and a score for the degree of confidence the system has in the detection. Importantly, there appears to be no attempt to determine which mentions co-refer, something that all the other APIs do; for Amazon, each mention is treated as a distinct and independent entity.

[10] See `https://console.bluemix.net/docs/services/natural-language-understanding/entity-types.html#entity-types-and-subtypes` for the complete list.

https://doi.org/10.1017/S1351324918000013 Published online by Cambridge University Press

Amazon Comprehend output format:

```
{'Entities': [{'BeginOffset': 183,
               'EndOffset': 194,
               'Score': 0.9975835680961609,
               'Text': 'New Zealand',
               'Type': 'LOCATION'},
               ...
}
```

Google Natural Language output format:

```
{'entities': [
  {'mentions': [
    {'text': {'beginOffset': 21, 'content': 'NEW ZEALAND'}, 'type': 'PROPER'},
    {'text': {'beginOffset': 183, 'content': 'New Zealand'}, 'type': 'PROPER'},
    {'text': {'beginOffset': 943, 'content': 'New Zealand'}, 'type': 'PROPER'}],
   'metadata': {'mid': '/m/02fbb5',
     'wikipedia_url': 'https://en.wikipedia.org/wiki/New_Zealand_national_cricket_team'},
     'name': 'NEW ZEALAND',
     'salience': 0.06472472,
     'type': 'LOCATION'},
...
}
```

IBM Natural Language output format:

```
{'entities': [{'count': 2,
               'disambiguation': {'subtype': ['Country']},
               'mentions': [{'location': [183, 194], 'text': 'New Zealand'},
                            {'location': [943, 954], 'text': 'New Zealand'}],
               'relevance': 0.295711,
               'text': 'New Zealand',
               'type': 'Location'},
               ...
```

Microsoft Azure output format:

```
{'entities': [{'matches': [{'entries': [{'offset': 21}], 'text': 'NEW ZEALAND'},
                           {'entries': [{'offset': 189}, {'offset': 989}],
                            'text': 'New Zealand'}],
               'name': 'New Zealand national cricket team',
               'score': 0.981,
               'wikipediaId': 'New Zealand national cricket team'},
               ...
```

Fig. 1. API results.

Google returns an array of entities with the individual mentions that are assumed to be references to each. The entity types are those shown in Table 1. Each entity has an overall salience within the document; mention types indicate whether the mention is a proper noun or a common noun. Where possible, each entity is also associated with a Wikipedia URL and a Knowledge Graph MID.[11] It's also possible to request the overall sentiment expressed towards the entity in the document.

IBM's output is similar to Google's, with each entity detected being associated with a set of mentions and their locations. If available, a DBpedia link for the entity

[11] See `https://developers.google.com/knowledge-graph/`.

https://doi.org/10.1017/S1351324918000013 Published online by Cambridge University Press

Table 2. *Results on the CoNLL shared task data; all values are percentages*

	Amazon comprehend			Google NL			IBM NL		
	Prec'n	Recall	$F_{\beta=1}$	Prec'n	Recall	$F_{\beta=1}$	Prec'n	Recall	$F_{\beta=1}$
LOC	**76.13**	72.66	74.36	58.81	**86.45**	70.00	70.17	86.15	**77.34**
MISC	**58.40**	10.40	17.65	36.76	**19.37**	**25.37**	2.08	0.14	0.27
ORG	**74.72**	**59.24**	**66.08**	68.03	48.16	56.40	69.86	27.63	39.60
PER	**87.14**	82.99	**85.02**	82.45	**83.36**	82.90	73.13	76.07	74.57
Overall	**78.95**	63.93	**70.65**	66.15	**65.97**	66.06	70.51	55.36	62.03

is provided.[12] As noted earlier, sentiment and emotion towards the entity can also be requested. A maximum of 250 entities are returned for a given input text.

Microsoft also provides an array of entities with their associated mentions in the text, tied together via a representative name. A Wikipedia ID is provided if one can be found, but as noted above, no type information is provided.

Figure 1 shows the results from each API for a particular named entity, in the context of a report on cricket match.[13] The entity referred to here is the New Zealand cricket team. The phrases *New Zealand* and *cricket* do not appear adjacent anywhere in the text, but both Google and Microsoft successfully resolve the entity to the cricket team rather than the country.

5 How well do they work?

The results shown in Table 2 are derived from running each of the APIs on the publicly available CoNLL 2003 shared task data, drawn from the Reuters Corpus.[14] There are a variety of issues here: the different APIs use different tokenisation algorithms, which sometimes produce results inconsistent with the gold standard (which doesn't mean they are wrong—sometimes the CoNLL annotations appear to be incorrect); and there are alternative ways you might map each API's entity types into the CoNLL entity types. It also probably wouldn't be fair to compare these numbers with those generated for the participants in the shared task, since the latter systems were presumably tailored to the Reuters training data, whereas that's unlikely to be true of the services reviewed here.[15] So, take these numbers with a grain of salt; still, I think it's interesting to see how the three systems compare with each other on what you might consider a relatively neutral dataset.[16]

[12] Strangely, as the example in Figure 1 shows, the API fails to find the DBpedia resource for New Zealand.
[13] This is Reuters file 239046newsML.
[14] See `https://www.clips.uantwerpen.be/conll2003/ner/`.
[15] IBM's poor performance on the MISC category appears to be largely due to it failing to recognise any nationality adjectives, like *Croat* and *German*, which CoNLL tags as MISC.
[16] Again, the Microsoft results are not represented here since these do not contain type information for the recognised entities.

https://doi.org/10.1017/S1351324918000013 Published online by Cambridge University Press

6 Pricing

As is often the case with SaaS products in a competitive marketplace, comparing pricing is difficult: pricing varies across the providers by the particular analytics service used, by the sizing adopted for text units, and by tiering splits. All four providers have either a free tier or a free trial period so you can try them out. For production usage, Amazon's unit size for named entity recognition is 100 characters, and is priced at $0.0001 per unit for up to 10 M units, discounted thereafter;[17] Google's unit size, called a text record, is 1,000 characters, with a $0.001 charge per text unit up to 1 M units, discounted thereafter;[18] and IBM charges in terms of NLU units, where a unit is 10 K characters × number of features requested (the entity recognition results shown here count as a single feature), costed at $0.003 per NLU item up to 250 K items per month.[19] I couldn't work out the pricing for the Microsoft offering; my best guess is it's free if you use some other services, but it's hard to tell. This may be due to the service currently being categorised as being in Preview mode.

Based on these numbers, if you wanted to process 100,000 documents of 10 KB in size each month, Amazon and Google would both charge you $1,000, but IBM would charge you only $300. But the different approaches to unit size and different tiering splits mean that the best deal will depend on your specific numbers.

7 And the winner is ...

Well, it all depends.

- If you care about the types of the entities detected, then Microsoft's Entity Linking API is a non-starter, since it doesn't (currently, at least) provide that information. When the API provides a Wikipedia link, you might be able to work out the type, but, as a good friend of mine likes to say, life's too short for tight shoes. Microsoft: 👎
- If you care about coralling the different references to the same entity within a document, then you don't want the Amazon API, since this treats every mention as independent. Amazon: 👎
- If you also want to make use of parsing results, you're limited to Google (for dependency parsing) and Microsoft (for constituency parsing); if you're just after semantic roles and relations, without access to a full syntactic analysis, IBM will do.[20] Google and Microsoft: 👍
- To the extent that the results of the evaluation on the CoNLL data presented above are at all representative, you might make a choice dependent upon the metric and entity types that are important to you. IBM's entity type

[17] See `https://aws.amazon.com/comprehend/pricing/`.
[18] See `https://cloud.google.com/natural-language/pricing`.
[19] See `https://www.ibm.com/watson/services/natural-language-understanding/`.
[20] Of course, you could use different providers for different functionalities, but that's a risky road; for example, as noted earlier, tokenisation isn't handled the same way across the four providers, so you might hit alignment problems if you try a mix-and-match approach.

https://doi.org/10.1017/S1351324918000013 Published online by Cambridge University Press

extensibility may be a trump card here: although it performed less well than the others with default settings, the ability to extend the set of entities recognised provides a way around this; so IBM: 👍

- As we saw in the *New Zealand* example discussed above, Google and Microsoft both do a good job of resolving an ambiguous reference to the correct context-dependent entity, whereas IBM does not.[21] Google and Microsoft: 👍
- The IBM API provides a number of 'management features' not provided by the others, such as text cleaning and metadata extraction. IBM: 👍
- Your mileage may vary, but I found the Amazon and IBM documentation easier to find things in. Google's documentation seems to be out of step in some places with the actual API, but bear in mind that it is classed as a beta release, so it may be changing more frequently than a stable release. Similarly, Microsoft's documentation seems rather limited, but again that might be accounted for by the Preview nature of the offering. Amazon and IBM: 👍

So, obviously, it all depends on your specific requirements.

But so far we've only been talking about the Goliaths; what about the smaller players? For the big players, it's possible that it's more important to have *a* text analytics offering than it is to have a *good* text analytics offering; maybe there are buying decisions where the suits just want to know if the provider ticks the text analytics box. But that doesn't wash for companies whose sole focus is text analytics. So how do they compare? We'll look at those in the next post—watch out for Text Analytics APIs, Part 2.

[21] Again, Amazon does not do any entity linking at all.

https://doi.org/10.1017/S1351324918000013 Published online by Cambridge University Press