Microsoft Azure

Tech Review

Text Analytics in Microsoft Azure

Seungill Kim

What is Text Analytics in Azure?

According to the Microsoft API document (https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/), Text Analytics is one of the Azure Cognitive services offered by Microsoft, which is a cloud-based service that provides natural language processing over raw text.

As well as Text Analytics, Azure Cognitive services provide a comprehensive service to build cognitive intelligence into your applications regarding vision, speech, language, decision, search. Text Analytics is classified as a subset of Language API. (https://docs.microsoft.com/en-us/azure/cognitive-services/what-are-cognitive-services)

Technically, that service is a pre-trained text-mining AI service that uncovers insights such as sentiment analysis, entities, relations, and key phrases in unstructured text. Since the model of the service is pre-trained, it is not allowed for users to strengthen the model. The only operations available on uploaded data are scoring, key phrase extraction, and language detection. (https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/text-analytics-resource-faq)

Therefore, that server is a black-box model. Users can only input data and get the result without any knowledge of how it works.

History of Text Analytics Service – The Microsoft cloud service strategy

Microsoft claims Text Analytics Service is based on Microsoft proprietary technology. Therefore, they do not open how it works, but we can speculate a part of the structure through the history of the service.

The history of Text Analytics Service

Text Analytics was a module in the Azure Machine Learning service in 2016. You can still see the early version of the Text Analytics module in the Azure AI Gallery. (https://gallery.azure.ai/Collection/Text-Analytics-Modules-in-Azure-Machine-Learning-1)

At that time, Text Analytics was one of the machine learning models in Microsoft Azure ML. Therefore users could train the machine to learn the language with their own strategy. However, Microsoft decided that the model and pre-trained data set are mature enough to commercialize for general use as a black-box model.

The Below figure is an overview of the Microsoft Azure NLP solution. Microsoft classifies Cognitive Services as a pre-built AI. Azure Cognitive Service, including Text Analytics, is a pre-trained intelligent service, therefore it is not allowed for users to customize or strengthen the model. The scope of the service is the service API to build applications only. (https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?context=azure%2fmachine-learning%2fstudio%2fcontext%2fml-context)

Furthermore, Microsoft recommends using SparkML or Databricks ML under Azure if users want to build the dedicated learning model. (https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/natural-language-processing) This shows the productizing strategy of Microsoft. (fully open – wait until the maturity – close for productization)

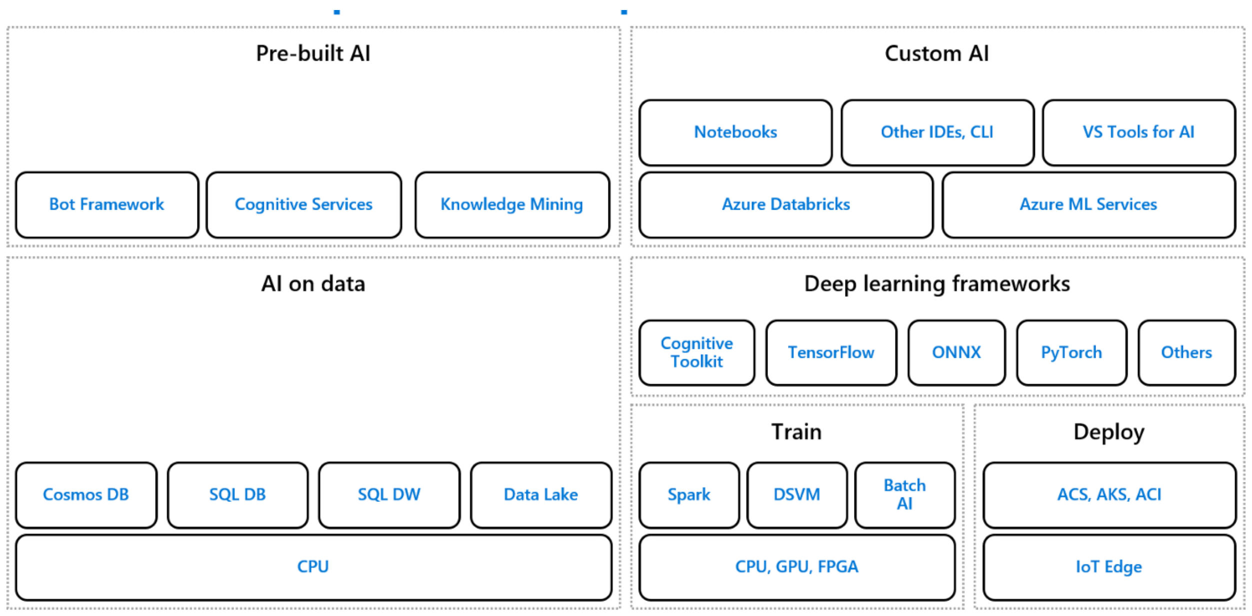

Figure 1. An overview of Microsoft NLP Solutions
(Re-citied from https://medium.com/analytics-vidhya/an-overview-of-microsoft-azure-nlp-solutions-b39dff61cee8)

Then, Is the existing Text Analytics module closed?

No. Although Microsoft classifies the "Text Analytics Service" as a black-box, pre-built AI service, most functionalities of the existing "Text Analytics Module" are still offered in the Azure Machine Learning Designer tool.

Instead of the separated module, those functionalities are fully integrated as designer learning functions. For example, the language learning through BERT, Word2Vec, and GloVe models are now the automated machine learning and Convert Word to Vector module in the designer tool.

(https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-features
https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/convert-word-to-vector)

We would take a look at the technical structure of the service considering those uniquenesses and complications. We suppose the current Text Analytics Service is still based on the technologies used in the Text Analytics "Module". However, it is not surprising that Microsoft modified the model and approach of the service. They claim the service is a pre-built AI based on Microsoft proprietary technologies.

Text Analytics Techniques

Microsoft defines text analytics as a process where an artificial intelligence (AI) algorithm, running on a computer, evaluates these same attributes in text, to determine specific insights. (https://docs.microsoft.com/en-us/learn/modules/analyze-text-with-text-analytics-service/1-introduction)

For evaluation and determination, there are some common techniques, including

- Statistical analysis: Removing common stop words and performing frequency analysis of the remaining words
- N-grams frequency analysis: Extending the analysis to more than two-term phrases
- Stemming or lemmatization: Normalization of words before counting them. (interpreting terms of the same stem. For example, “go”, “goes”, and “went” should be counted as the same word.
- Linguistic structure rules: Breaking down sentences into tree-like structures (as we learned in week 1 to explain why text analytics are complicated job) For example, “good person” is classified as a noun phrase rather than an independent combination of “good” and “person”.
- Encoding words or terms as numeric features: This technique is generally used for sentiment analysis to score the degree of positiveness and negativeness.
- Vectorized models: The process maps words into vector space. A vector space represents each word by a vector of real numbers. Also, it allows words with similar meanings have similar representations. Therefore, vectorized models can capture semantic relationships between words by assigning them.

Using those analytics techniques, Microsoft claims the Text Analytics cognitive service can help simplify application development by using pre-trained models that can:

- Determine the language of a document or text (for example, French or English).
- Perform sentiment analysis on text to determine a positive or negative sentiment.
- Extract key phrases from the text that might indicate its main talking points.
- Identify and categorize entities in the text. Entities can be people, places, organizations, or even everyday items such as dates, times, quantities, and so on.

(https://docs.microsoft.com/en-us/learn/modules/analyze-text-with-text-analytics-service/1-introduction)

The technical architecture of the service

Azure Cognitive Service, including Text Analytics, is a pre-trained intelligent service, therefore it is not allowed for users to customize or strengthen the model. Thus, underlying technologies are Microsoft proprietary.

However, as explained above, the current Text Analytics Service started from the Text Analytics module in Azure ML(Machine Learning), and Microsoft opened some of the underlying technology and model at that time.

Below is the scenario and model table called “NLP recipe” from Microsoft. We can speculate the technical architecture of the service based on that.

But, that information is as of 2019, and it is not surprising that Microsoft changed the model and approach to improve their service. They claim the service is a pre-built AI based on Microsoft proprietary technologies. Two years is enough time in the AI domain to create a new innovative model.

Scenario	**Models**	**Description**	**Languages**
Text Classification	BERT, DistillBERT, XLNet, RoBERTa, ALBERT, XLM	Text classification is a supervised learning method of learning and predicting the category or the class of a document given its text content.	English, Chinese, Hindi, Arabic, German, French, Japanese, Spanish, Dutch
Named Entity Recognition	BERT	Named entity recognition (NER) is the task of classifying words or key phrases of a text into predefined entities of interest.	English
Text Summarization	BERTSumExt BERTSumAbs UniLM (s2s-ft) MiniLM	Text summarization is a language generation task of summarizing the input text into a shorter paragraph of text.	English
Entailment	BERT, XLNet, RoBERTa	Textual entailment is the task of classifying the binary relation between two natural-language texts, *text* and *hypothesis*, to determine if the *text* agrees with the *hypothesis* or not.	English

Scenario	Models	Description	Languages
Question Answering	BiDAF, BERT, XLNet	Question answering (QA) is the task of retrieving or generating a valid answer for a given query in natural language, provided with a passage related to the query.	English
Sentence Similarity	BERT, GenSen	Sentence similarity is the process of computing a similarity score given a pair of text documents.	English
Embeddings	Word2Vec fastText GloVe	Embedding is the process of converting a word or a piece of text to a continuous vector space of real numbers, usually in low dimensions.	English
Sentiment Analysis	Dependency Parser GloVe	Provides an example of train and use Aspect Based Sentiment Analysis with Azure ML and Intel NLP Architect.	English

Table 2. NLP recipes (https://github.com/microsoft/nlp-recipes)

The BERT model in Azure

The table above shows that the model heavily depends on the BERT(and its variants) and vectorization models such as GloVe, Word2Vec models. Although each model is an independent technical review topic, we still have a brief overview to explain the implementation of the models in Microsoft Azure.

A brief overview of the BERT model

According to Wikipedia, The Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. (https://en.wikipedia.org/wiki/BERT_(language_model))

This model is an improvement of the Transformers-based model. While other NLP models depend on sequence transduction for text analysis, Transformers applies a self-attention mechanism that directly models relationships between all words in a sentence, regardless of their relevant position. It turned that it was more effective to focus only on the relationship(attention), and the Transformer got immediate success on the NLP domain.

Based on the Transformer, the BERT model applies bidirectional training. Previous models had conducted the training through a text sequence either from left to right or combined left-to-right and right-to-left training. In contrast, the BERT model takes both the previous and next word tokens into account simultaneously.

That is the key improvement of the BERT over other models. For example, when speculating the meaning of "ring" in the sentence, the BERT model evaluates both previous and next context. For example, there are two sentences,

- Here is a new ring **for your mobile**.
- Here is a new ring **for your girlfriend**.

Unidirectional contextual models would represent "ring" based on "Here is a new" but not the next context. However, the BERT model evaluates "ring" using both previous("Here is a") and next context("for your mobile" and "for your girlfriend"), starting from the very bottom of a deep neural network, making it deeply bidirectional.

Below articles and tutorials will help for the in-depth understanding of the BERT model.

https://arxiv.org/abs/1706.03762

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

https://jalammar.github.io/illustrated-bert/

https://towardsml.com/2019/09/17/bert-explained-a-complete-guide-with-theory-and-tutorial/

https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

The BERT model implementation in Microsoft Azure

Basically, the BERT model is implemented as it is in Azure. But, the Microsoft NLP team tried to improve several challenges during the implementation of the BERT model, especially from the training performance perspective.

Pretraining a BERT is quite challenging. Instead of training it from scratch, most developers start the development based on the pre-trained with a standard corpus (such as Wikipedia). This strategy works well if the final model is also on a similar corpus in the pre-train stage.

However, the general model does not satisfy all. The model training is unavoidable if the problem involves a specialized corpus or developers want an advanced language representation beyond BERT's accuracy. But, this requires building a very complicated distributed infrastructure environment with a lot of servers and GPUs.

To address this issue, Microsoft implemented a workflow for pre-training BERT-large models. That offered flexibility in underlying machine learning infrastructure by simplifying the resource allocation. Moreover, there are several developer support tools such as a PyTorch implementation of the BERT model from Hugging Face repo, raw pre-processed English Wikipedia datasets, and data preparation scripts when Text Analytics was a part of Azure ML.

(https://github.com/Microsoft/AzureML-BERT
https://azure.microsoft.com/en-us/blog/microsoft-makes-it-easier-to-build-popular-language-representation-model-bert-at-large-scale/)

Although the Text Analytics module is now commercialized as a service, those functionalities are still offered in Azure Machine Learning. Instead of a separated module, the BERT is used in the featurization layer of Azure automated machine learning (AutoML). Featurization means collective techniques for data-scaling and normalization in Azure Machine learning. (https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-features)

The vectorization model in Azure

A brief explanation of the vectorization model

Another important Azure Text Analytics Service model is the vectorization model such as GloVe, Word2Vec, and fastText. Basically, that is based on the vector space model we learned. The relationship between words is derived using the cosine value between words. As well as the GloVe model, other vector-based language learning models such as Word2Vec and fastText have the same algorithm for relationship calculation.

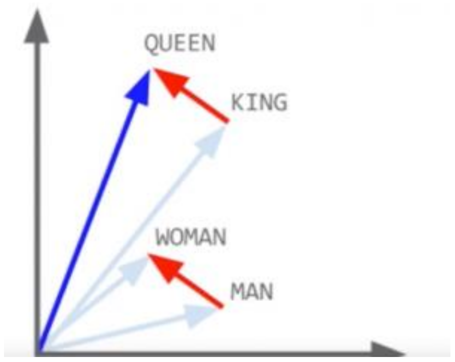

Figure 2. Words relationship in vector-based language learning models

(https://machinelearninginterview.com/topics/natural-language-processing/what-is-the-difference-between-Word2Vec-and-glove)

But there are differences amongst those models. First, the GloVe model (as the name derived from **Glo**bal **Ve**ctor) is based on global word-to-word co-occurrence counts leveraging the "entire" corpus and global matrix of co-occurrence information which is words by context. According to Stanford NLP, GloVe is a log-bilinear model with a weighted least-squares objective which conducts training on aggregated global word-word co-occurrence statistics from the corpus. (https://nlp.stanford.edu/projects/glove/)

On the other hand, Word2Vec and fastText model depend on the local context (neighboring words). The Word2Vec model treats each word as an atomic entity and generates a vector for each word, whereas the fastText treats each word as composed of character n-grams. Therefore the vector for a word is made of the sum of this character n-grams. That is the difference between Word2Vec and fastText.

Below articles and tutorials will help for the in-depth understanding of the vectorization model.

https://nlp.stanford.edu/projects/glove/
https://nlp.stanford.edu/pubs/glove.pdf
https://github.com/stanfordnlp/GloVe
https://radimrehurek.com/gensim/models/fasttext.html
https://arxiv.org/pdf/1411.2738.pdf
https://arxiv.org/pdf/1607.01759.pdf
https://machinelearninginterview.com/topics/natural-language-processing/what-is-the-difference-between-Word2Vec-and-glove/
https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010
https://medium.com/swlh/a-quick-overview-of-the-main-difference-between-Word2Vec-and-fasttext-b9d3f6e274e9
https://medium.com/analytics-vidhya/word-embeddings-in-nlp-Word2Vec-glove-fasttext-24d4d4286a73

The vectorization model implementation in Microsoft

Similar to the BERT model, vectorization models are integrated into the Azure Machine Learning. Now they are underlying technologies in the Covert Word to Vector module in Azure Machine Learning designer. All models are implemented as they are. Unlike the BERT model, however, all three vectorization models are used to implement one functionality in Azure Machine Learning. You can see more information how they were implemented on Microsoft documentation.

https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/convert-word-to-vector

Moreover, the Word2Vec model is implemented in the SparkML and DatabrickML services in Azure as well. As I described earlier, Microsoft recommends using both services to implement a dedicated NLP algorithm. Both are implemented as they are in Azure, and Microsoft offers API to use them. The below links will help to understand how they work in Azure. (For your information, Microsoft is a noted investor of Databrick)

https://github.com/microsoft/SynapseML
https://docs.microsoft.com/ko-kr/rest/api/databricks/

The future of Microsoft NLP

As described above, the current Microsoft NLP model is a combination of various models and heavily depends on BERT and vectorization models. But recently, Microsoft teamed up with OpenAI and exclusively licensed GPT-3 language model. (https://blogs.microsoft.com/blog/2020/09/22/microsoft-teams-up-with-openai-to-exclusively-license-gpt-3-language-model/)

I would not explain the technical detail of the GPT-3 model. But, Microsoft is preparing to adopt the GPT-3 to Azure service and announced the first product features powered by GPT-3 in May. (https://blogs.microsoft.com/ai/from-conversation-to-code-microsoft-introduces-its-first-product-features-powered-by-gpt-3/) Interestingly, this is an integration with the Power App rather than a machine learning model or functionality.

This announcement shows that Microsoft is interested in the integration of their NLP technology with other Microsoft products rather than machine learning itself. As I have explained, the Text Analytics Service started from one of the machine learning modules but now services as a black-box model to the general developers. In my opinion, Microsoft decided that the quality of NLP is mature enough for the general public beyond the IT professional and developer group. Therefore, it appears that their focus has shifted from the model itself to how to make ease of use and adopt. Thus, the GPT-3 model is expected to appear as an applied service rather than the machine learning domain.

Conclusion

Text Analytics is one of the Azure Cognitive services offered by Microsoft, a cloud-based service that provides natural language processing over raw text. In the past, that was one of the Azure ML modules that allowed language learning by the developer. However, Microsoft decided that the model and pre-trained data set are mature enough to commercialize, so they sell as a full-functioning black-box model currently. But those functionalities are still offered in Azure Machine Learning as integrated functions in the designer tool.

Technically, the underlying model of Text Analytics is a combination of various models and heavily depends on BERT and vectorization models. But recently, Microsoft teamed up with OpenAI and exclusively licensed GPT-3 language model. Therefore, we can expect that the current BERT model would be changed to the GPT-3. But, it seems that Microsoft shifted their focus from models themselves to how to make ease of use and adopt. Therefore, the GPT-3 model is expected to appear as an applied service rather than the machine learning domain.