GENERAL ASSEMBLY

NATURAL LANGUAGE PROCESSING

Joseph Nelson, Data Science Immersive

AGENDA

- Introduction to NLP / Use Cases
- Bag of Words, Tokenization
- Count Vectorization, Ngram, Stopword removal
- Hashing Vectorization
- TD-IDF

- Next class: Stemming, lemmatization, POS tagging, Textblob

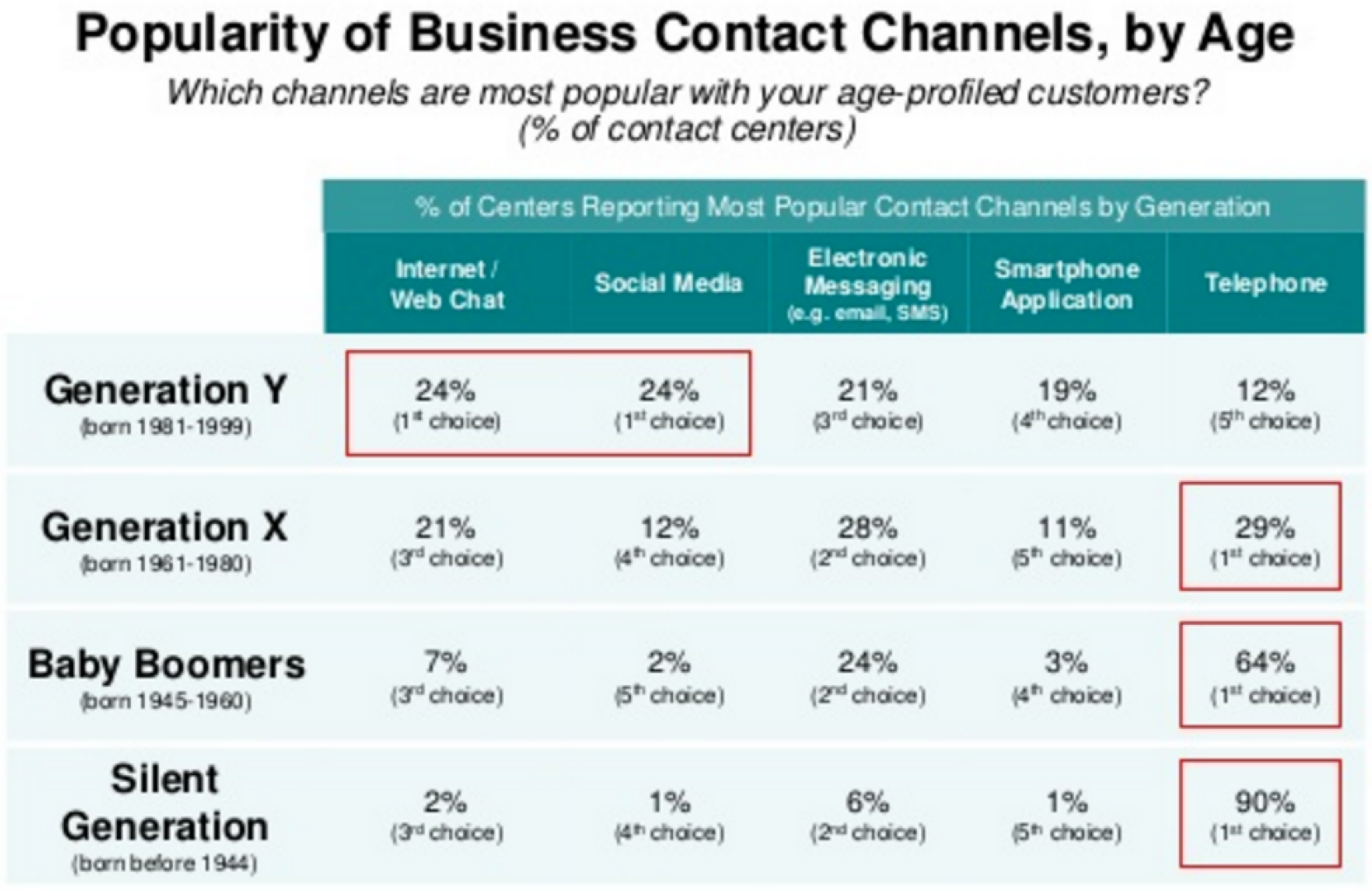
WHAT IS NLP?

- Natural language processing (#NLProc) is the act of extracting features and insights from unstructured text
- Cooler definition: enabling computers to understand language the same way people do

WHERE IS IT USED?

- Informational retrieval (google.com)
- Information Extraction (events from Gmail)
- Machine Translation (Google translate)
- Text simplification (Rewordify)
- Predictive text input (Autocrrect)
- Sentiment Analysis (How angry ARE Donald Trump’s tweets?)
- Speech Recognition and generation (text-to-speech)
- Question answering (IBM Watson)

AND IT'S GROWING IN IMPORTANCE…

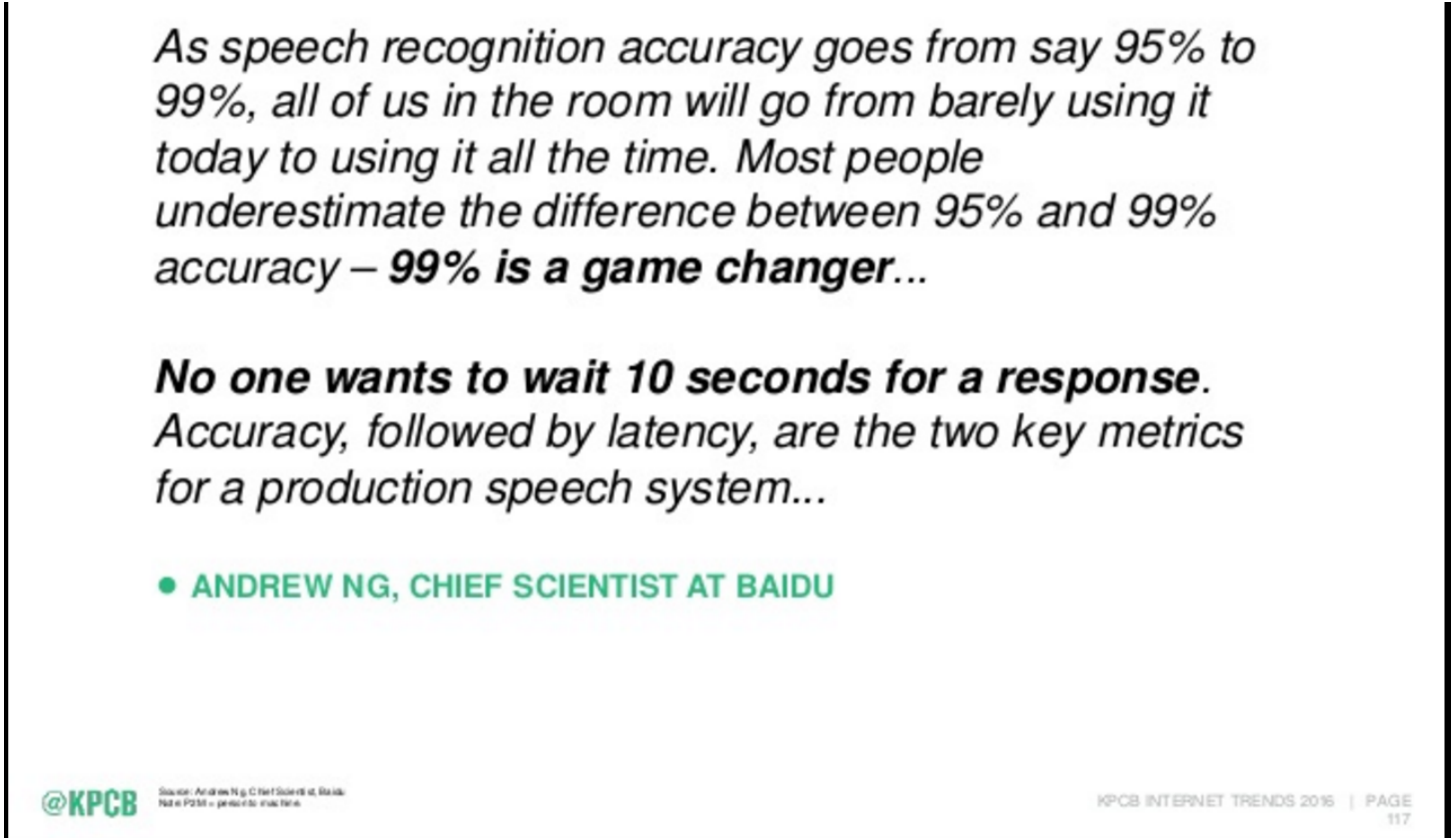


Popularity of Business Contact Channels, by Age

Which channels are most popular with your age-profiled customers?
(% of contact centers)

	% of Centers Reporting Most Popular Contact Channels by Generation				
	Internet / Web Chat	Social Media	Electronic Messaging (e.g. email, SMS)	Smartphone Application	Telephone
Generation Y (born 1981-1999)	24% (1st choice)	24% (1st choice)	21% (3rd choice)	19% (4th choice)	12% (5th choice)
Generation X (born 1961-1980)	21% (3rd choice)	12% (4th choice)	28% (2nd choice)	11% (5th choice)	29% (1st choice)
Baby Boomers (born 1945-1960)	7% (3rd choice)	2% (5th choice)	24% (2nd choice)	3% (4th choice)	64% (1st choice)
Silent Generation (born before 1944)	2% (3rd choice)	1% (4th choice)	6% (2nd choice)	1% (5th choice)	90% (1st choice)

AND IT’S GROWING IN IMPORTANCE…



*As speech recognition accuracy goes from say 95% to 99%, all of us in the room will go from barely using it today to using it all the time. Most people underestimate the difference between 95% and 99% accuracy – **99% is a game changer**...*

***No one wants to wait 10 seconds for a response**. Accuracy, followed by latency, are the two key metrics for a production speech system...*

- ANDREW NG, CHIEF SCIENTIST AT BAIDU

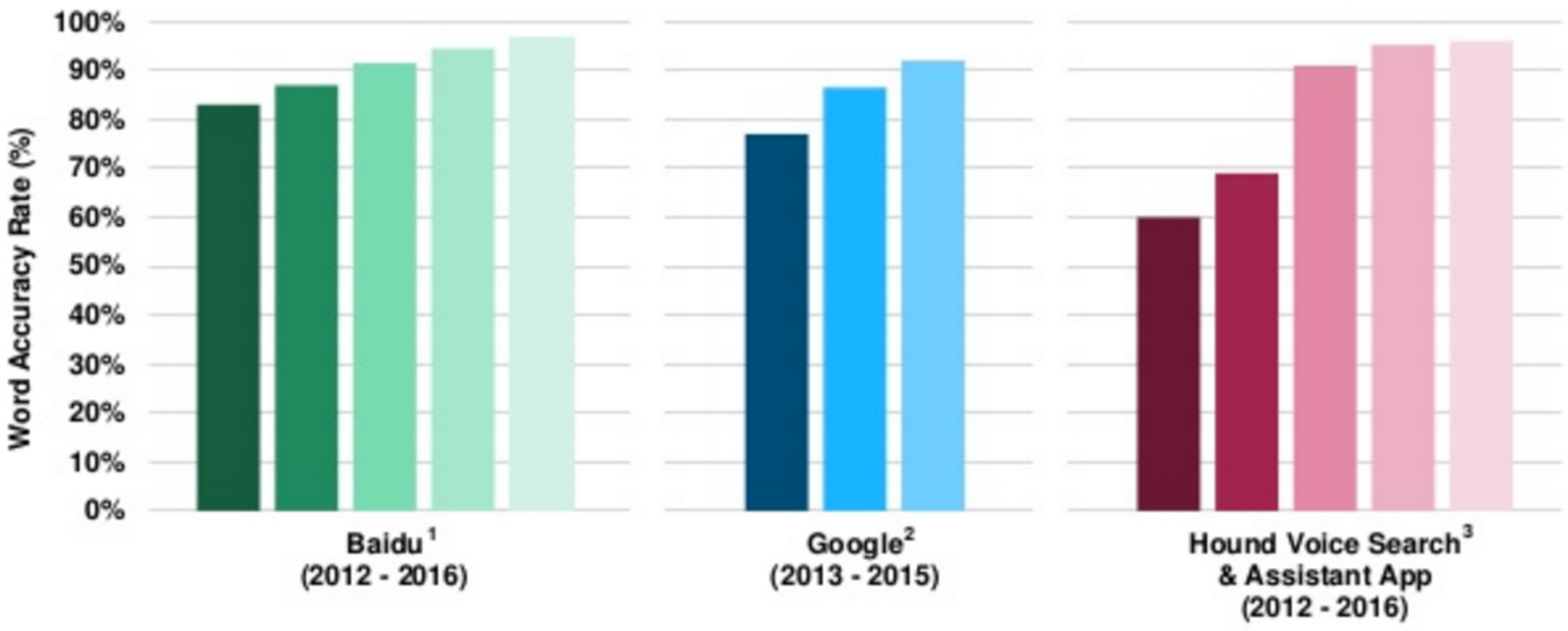
@KPCB

Source: Andrew Ng, Chief Scientist, Baidu.
Note: P2M = person to machine.

AND IT’S GROWING IN IMPORTANCE…

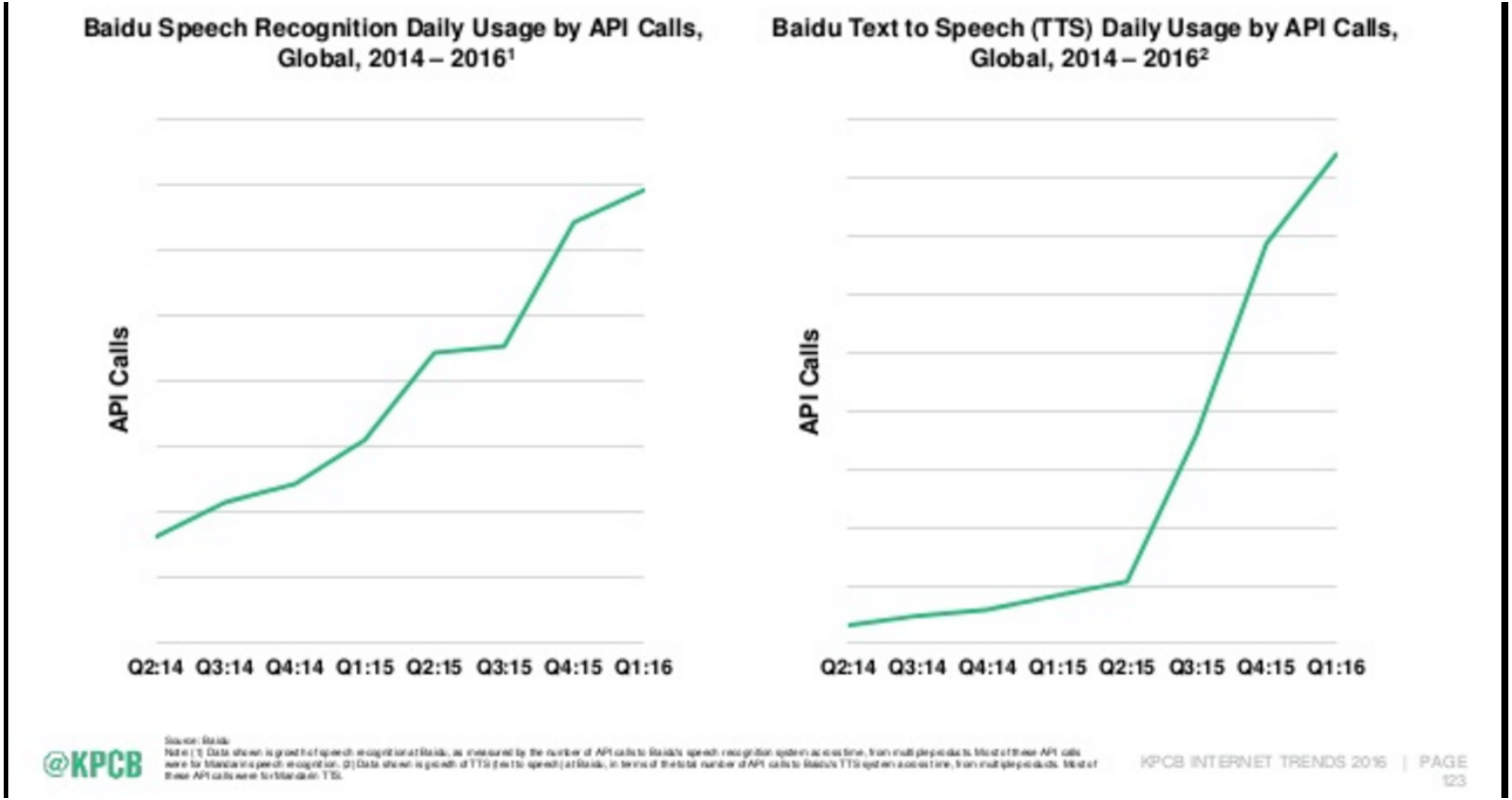
Word Accuracy Rates by Platform*, 2012 – 2016

*Word accuracy rate definitions are unique to each company…see footnotes for more details



Source: Baidu, Google, VentureBeat, SoundHound
Note: *Word Error Rate (WER) definitions are specific to each company. Word accuracy rate = 1 - WER. (1) Data shown is word accuracy rate on Mandarin speech recognition on one of Baidu's speech tasks. Real world mobile phone speech data is very noisy and hard for humans to transcribe. A 3.5% WER is better than what most native speakers can accomplish on this task. WER across different datasets and languages are generally not comparable. (2) Data as of 5/15 and refers to recognition accuracy for English language. Word error rate is evaluated using real world search data which is extremely diverse and more error prone than typical human dialogue. (3) Data as of 1/16 and refers to recognition accuracy for English language. Word accuracy rate based on data collected from thousands of speakers and real world queries with noise and accents.

IN SUM, VOICE IS BECOMING ITS OWN COMPUTING INTERFACT



Baidu Speech Recognition Daily Usage by API Calls, Global, 2014 – 2016[1]

Baidu Text to Speech (TTS) Daily Usage by API Calls, Global, 2014 – 2016[2]

Source: Baidu

Note: (1) Data shown is growth of speech recognition at Baidu, as measured by the number of API calls to Baidu's speech recognition system across time, from multiple products. Most of these API calls were for Mandarin speech recognition. (2) Data shown is growth of TTS (text to speech) at Baidu, in terms of the total number of API calls to Baidu's TTS system across time, from multiple products. Most of these API calls were for Mandarin TTS.

KPCB INTERNET TRENDS 2016 | PAGE 123

BUT NLP IS HARD

- 100% real outputs trying to predict headlines!!
- Hospitals are Sued by 7 Foot Doctors

BUT NLP IS HARD

- 100% real outputs trying to predict headlines!!

- Hospitals are Sued by 7 Foot Doctors
- Juvenile Court to Try Shooting Defendant Local

BUT NLP IS HARD

- 100% real outputs trying to predict headlines!!

- Hospitals are Sued by 7 Foot Doctors
- Juvenile Court to Try Shooting Defendant Local
- High School Dropouts Cut in Half

EXTRACTING INSIGHTS FROM UNSTRUCTURED TEXT

- Your goal: build a model that identifies a given LinkedIn message as spam or ham.

- Your data set (corpus) is the following slide.

- Deliverable: a step-by-step description of your method. Ready?

EXTRACTING INSIGHTS FROM…UNSTRUCTURED! TEXT

- Hello, I saw your contact information on LinkedIn. I have carefully read through your profile and you seem to have an outstanding personality. This is one major reason why I am in contact with you. My name is Mr. Valery Grayfer Chairman of the Board of Directors of PJSC "LUKOIL". I am 86 years old and I was diagnosed with cancer 2 years ago. I will be going in for an operation later this week. I decided to WILL/Donate the sum of 8,750,000.00 Euros(Eight Million Seven Hundred And Fifty Thousand Euros Only etc. etc.

- Hello, I am writing in regards to your application to the position of Data Scientist at Hooli X. We are pleased to inform you that you passed the first round of interviews and we would like to invite you for an on-site interview with our Senior Data Scientist Mr. John Smith. You will find attached to this message further information on date, time and location of the interview. Please let me know if I can be of any further assistance. Best Regards.

COUNT VECTORIZER

- You just performed a manual count vectorization process! Congrats. You are fully capable of being automated.

- SKLearn includes a method called “Count Vectorizer” that will return the number of occurrences of a given feature, where each feature is a word.

- Take a look at the SKLearn documentation. There’s one particular parameter we made be interested in discussing (right now): http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

COUNT VECTORIZER

- There are a few notable parameters we'll discuss: max_df, n_gram, stop_words

- What is max_df? Why would this matter?

COUNT VECTORIZER

- There are a few notable parameters we’ll discuss: max_df, n_gram, stop_words

- What is max_df? Why would this matter?

- This sets a maximum size for how many features we’ll include. This is significant because we may have documents of extreme size (imagine legal contracts or bills!). max_df removes features that do not meet a specific threshold of frequency, only keeping the most used features that are under this threshold.

COUNT VECTORIZER – NGRAM

- There are a few notable parameters we'll discuss: max_df, ngram_range, stop_words

- What is ngram_range? Why would this matter?

COUNT VECTORIZER – NGRAM

- There are a few notable parameters we’ll discuss: max_df, ngram_range, stop_words

- What is ngram_range? Why would this matter?

- https://books.google.com/ngrams/graph?content=data+science&year_start=1800&year_end=2000&corpus=15&smoothing=3&share=&direct_url=t1%3B%2Cdata%20science%3B%2Cc0

COUNT VECTORIZER – NGRAM

- There are a few notable parameters we'll discuss: max_df, ngram_range, stop_words

- What is ngram_range? Why would this matter?

- Ngrams are contiguous sequences of n items.
- Eg: ngram=2: ngrams are, are contiguous, contiguous sequences, sequences of, of n, n items

COUNT VECTORIZER – NGRAM

- There are a few notable parameters we'll discuss: max_df, ngram_range, stop_words

- What is ngram_range? **Why would this matter?**

- Ngrams are contiguous sequences of n items.
- Eg: ngram=2: ngrams are, are contiguous, contiguous sequences, sequences of, of n, n items

COUNT VECTORIZER – STOPWORDS

- There are a few notable parameters we'll discuss: max_df, ngram_range, stop_words

- What is stop_words? Why would this matter?

Stop words

From Wikipedia, the free encyclopedia

Not to be confused with Safeword.

COUNT VECTORIZER – STOPWORDS

- There are a few notable parameters we'll discuss: max_df, ngram_range, stop_words

- What is stop_words? Why would this matter?

- Stop words are words we filter out when processing text because they do not provide useful meaning. Including them adds unnecessary and irrelevant features.
- Eg "a, of, the, is, which"

HASHING VECTORIZER

- Two of the key limitations of count vectorizer are the size of features we must log, and the lack of ability to include out-of-corpus features.

- We use HashingVectorizer, which converts a collection of text documents to a matrix of occurrences, calculated with the hashing trick. Each word is mapped to a feature with the use of a hash function that converts it to a hash. If we encounter that word again in the text, it will be converted to the same hash, allowing us to count word occurrence without retaining a dictionary in memory.

- The main drawback of the this trick is that it's not possible to compute the inverse transform, and thus we lose information on what words the important features correspond to. The hash function employed is the signed 32-bit version of Murmurhash3.

TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY

- More interesting than stop-words is the tf-idf score. This tells us which words are most discriminating between documents. Words that occur a lot in one document but doesn't occur in many documents will tell you something special about the document.

- This relies on two stats: term frequency, and inverse document frequency.

TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY

- Term frequency:

- The raw frequency is how often term t exists in document d: tf(t,d)

- Inverse document frequency is whether a term is common or rare among all documents

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

- N: the number of documents
- The number of documents (d in corpus D) where term t appears (t in that doc, d)

TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY

- We, thus, multiply these terms:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

- Note that this value will always be greater than or equal to zero. tf(t,d) is (word/total words) in the document (bound [0,1]) and idf(t,D) is always greater than or equal to one as log(x/≤x) is ≥0.