IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

BUFFALO PATENTS, LLC,

Plaintiff,

v.

ZTE CORPORATION,

Defendant.

CIVIL ACTION NO. 6:22-cv-423

ORIGINAL COMPLAINT FOR
PATENT INFRINGEMENT

JURY TRIAL DEMANDED

ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Buffalo Patents, LLC ("Buffalo Patents" or "Plaintiff") files this original complaint against Defendant ZTE Corporation ("ZTE" or "Defendant"), alleging, based on its own knowledge as to itself and its own actions and based on information and belief as to all other matters, as follows:

PARTIES

1. Buffalo Patents is a limited liability company formed under the laws of the State of Texas, with its principal place of business at 1200 Silver Hill Dr., Austin, Texas, 78746.

2. ZTE Corporation is a Chinese corporation with a place of business at ZTE Plaza, Keji Road South, Hi-Tech Industrial Park, Nanshan District, Shenzhen, Guangdong Province, China 518057. ZTE may be served with process by serving the Texas Secretary of State, 1019 Brazos Street, Austin, Texas 78701, as its agent for service because it engages in business in Texas but has not designated or maintained a resident agent for service of process in Texas as required by statute. This action arises out of that business.

3. ZTE Corporation is the head of an interrelated group of companies that describes itself as a "world-leading listed integrated telecommunications equipment manufacturer in the world market and a provider of integrated global telecommunications solutions."[1]

4. ZTE and its affiliates, including ZTE (TX) Inc. and ZTE (USA) Inc., are part of the same corporate structure and distribution chain for the making, importing, offering to sell, selling, and using of the accused devices in the United States, including in the State of Texas generally and this judicial district in particular.

5. ZTE and its affiliates share the same management, common ownership, advertising platforms, facilities, distribution chains and platforms, and accused product lines and products involving related technologies.

6. Thus, ZTE and its affiliates operate as a unitary business venture and are jointly and severally liable for the acts of patent infringement alleged herein.

JURISDICTION AND VENUE

7. This is an action for infringement of United States patents arising under 35 U.S.C. §§ 271, 281, and 284–85, among others. This Court has subject matter jurisdiction of the action under 28 U.S.C. § 1331 and § 1338(a).

8. This Court has personal jurisdiction over ZTE pursuant to due process and/or the Texas Long Arm Statute because, *inter alia*, (i) ZTE has done and continues to do business in Texas; and (ii) ZTE has committed and continues to commit acts of patent infringement in the State of Texas, including making, using, offering to sell, and/or selling accused products in Texas, and/or importing accused products into Texas, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in Texas,

[1] ZTE Annual Report 2021, at 10, www.zte.com.cn/mediares/zte/Investor/20220311/E1.pdf.

and/or committing at least a portion of any other infringements alleged herein in Texas. In addition, or in the alternative, this Court has personal jurisdiction over ZTE pursuant to Fed. R. Civ. P. 4(k)(2).

9. Venue is proper as to Defendant ZTE, which is organized under the laws of a foreign jurisdiction. 28 U.S.C. § 1391(c)(3) provides that "a defendant not resident in the United States may be sued in any judicial district, and the joinder of such a defendant shall be disregarded in determining where the action may be brought with respect to other defendants." *See also In re HTC Corp.*, 889 F.3d 1349 (Fed. Cir. 2018).

BACKGROUND

10. The patents-in-suit broadly cover technology used in electronic devices commonly used today, such as mobile phones, tablets, and other devices. More particularly, the patents-in-suit describe key improvements to electronic devices in the areas of display technology and intelligent message recognition systems.

11. U.S. Patent No. 6,856,086 ("the '086 Patent") generally relates to the field of optical display devices. It discloses, *inter alia*, hybrid display devices that include a front panel, a back panel, and a light control material and methods for making hybrid display devices. In particular, the '086 Patent describes hybrid display devices that include rigid and flexible substrates. The patented technology is used in smartphones and other display devices, such as devices with flexible OLED displays.

12. The technology disclosed in the '086 Patent was developed by engineers at Avery Dennison Corporation in the late 1990s and early 2000s. Avery Dennison, a global Fortune 500 company, began its operations in the 1930s as the first self-adhesive label company. In the early 2000s—when the patent application that led to the '086 Patent was filed—Avery Dennison was involved, *inter alia*, in the development of specialty films and performance polymers. The

company continues to be known today as a global materials science and manufacturing company, specializing in the design and manufacture of labeling and functional materials.

13. The invention disclosed in the ’086 Patent has been cited during patent prosecution multiple times by leading technology companies, including 3M, Apple, Applied Materials, Asahi Glass Co. (Japan), BOE Technology Group (China), Eastman Kodak, General Electric, Hewlett Packard, and Sharp Corporation.

14. U.S. Patent Nos. 6,904,405 (“the ’405 Patent”) and 8,204,737 (“the ’737 Patent”) generally relate to the fields of speech and handwriting recognition technology. The patented technology is used today in connection with virtual keyboards on smartphones and other devices that include speech and handwriting conversion to text data. The ’405 Patent discloses, *inter alia*, using language models to improve conversion of speech and handwritten data into text data by training one of the language models. The ’737 Patent discloses, *inter alia*, converting speech and handwritten data into text data, where one of the language models is adjusted.

15. The inventor of the technology of these patents has multiple patents relating to the fields of speech and handwriting recognition, cryptography, digital signatures, and audio signal reconstruction. As an undergraduate at the University of Washington during the mid-1990s, he also invented a radio receiver technology with novel ways of tuning a radio among several channels.[2]

16. The inventions disclosed in the ’405 and ’737 patents have been cited during patent prosecution multiple times by many of the leading technology companies worldwide, including Alcatel-Lucent (later acquired by Nokia), Amazon, Apple, Cisco, Dell, Facebook, GM, Google, HP, Honda, Honeywell, IBM, Intel Lenovo, Microsoft, Mitsubishi, Motorola, Nuance,

[2] https://www.wrfseattle.org/story/edwin-a-suominen-engineer-and-inventor/

Palm (later acquired by HP), Panasonic, Philips, Qualcomm, Research in Motion (now Blackberry), Robert Bosch, Samsung, Siemens, Sony, SRI International, and the US Navy.

COUNT I

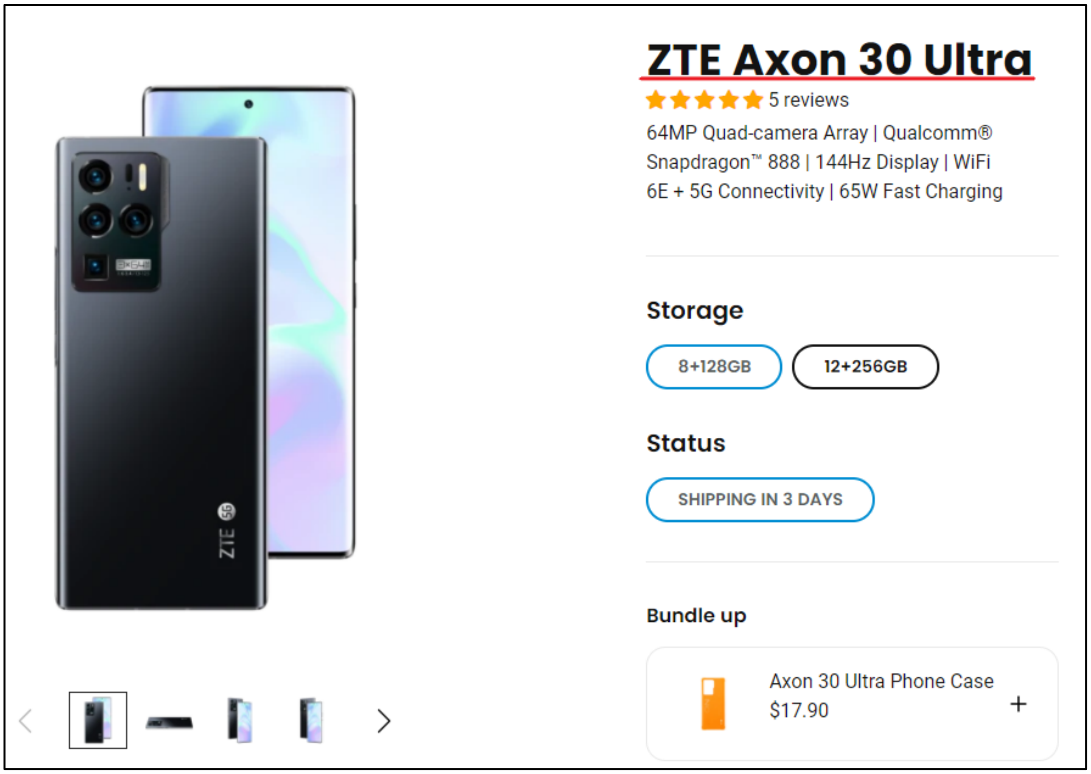
DIRECT INFRINGEMENT OF U.S. PATENT NO. 6,856,086

17. On February 15, 2005, the ’086 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Hybrid Display Device.”

18. Buffalo Patents is the owner of the ’086 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the ’086 Patent against infringers, and to collect damages for all relevant times.

19. ZTE made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, its ZTE Axon 30 Ultra smartphone and other products[3] that include curved displays or flexible OLED displays (“accused products”):

[3] *See, e.g.*, ZTE Axon 7, AXON 10 Pro, Axon 10S Pro, Axon 11, Axon 20 5G, Axon 30, Axon 30 Pro+, Blade 20 Pro, Nubia Z20, Nubia Z30 Pro, Nubia Z40 Pro, Nubia Alpha Smart Watch, Nubia Watch, Nubia Red Magic 6 Pro, Nubia Red Magic 6, Nubia Red Magic 5S, Nubia Red Magic 5G, Cymbal Z-320, Axon 7 Mini, etc.



ZTE Axon 30 Ultra

★★★★★ 5 reviews

64MP Quad-camera Array | Qualcomm® Snapdragon™ 888 | 144Hz Display | WiFi 6E + 5G Connectivity | 65W Fast Charging

Storage

8+128GB 12+256GB

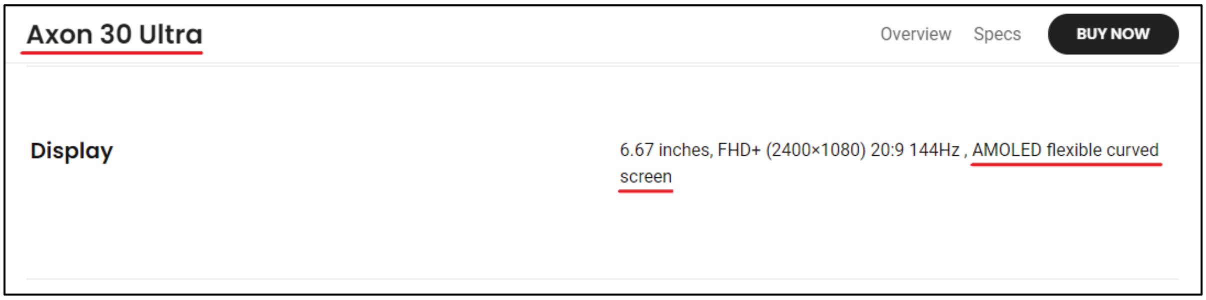
Status

SHIPPING IN 3 DAYS

Bundle up

Axon 30 Ultra Phone Case
$17.90 +

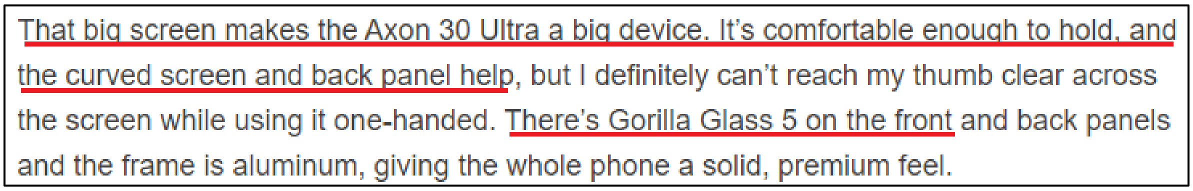
Source: https://na.ztedevices.com/products/axon-30-ultra



Axon 30 Ultra

Display	6.67 inches, FHD+ (2400×1080) 20:9 144Hz , AMOLED flexible curved screen

Source: https://ztedevices.com/en-us/axon-30-ultra-specs/



That big screen makes the Axon 30 Ultra a big device. It's comfortable enough to hold, and the curved screen and back panel help, but I definitely can't reach my thumb clear across the screen while using it one-handed. There's Gorilla Glass 5 on the front and back panels and the frame is aluminum, giving the whole phone a solid, premium feel.

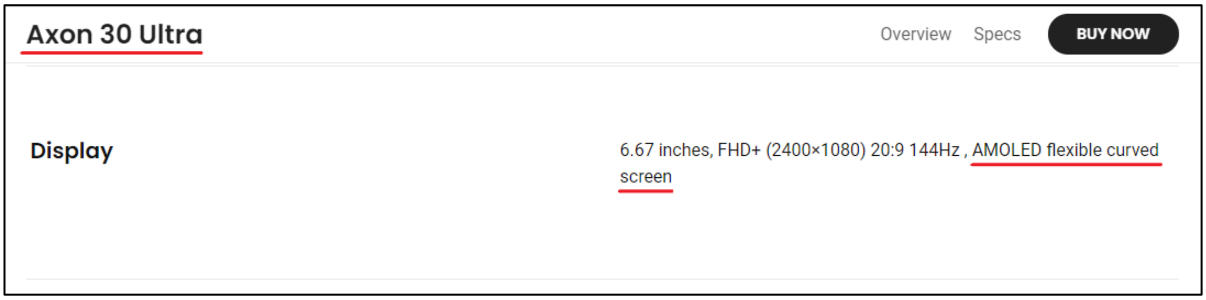
Source: https://www.theverge.com/22458629/zte-axon-30-ultra-review-screen-price-specs

20. By doing so, ZTE has directly infringed (literally and/or under the doctrine of equivalents) at least Claim 1 of the '086 Patent. ZTE's infringement in this regard is ongoing.

21. The ZTE Axon 30 Ultra smartphone is an exemplary accused product.

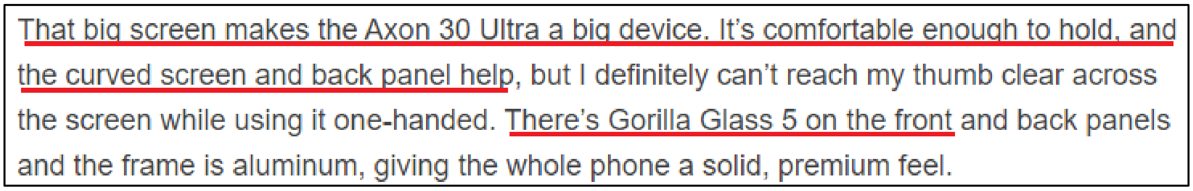
22. The ZTE Axon 30 Ultra smartphone is a display device comprising a front panel and a back panel with a light control material therebetween. One of the panels has a rigid substrate and the other of the panels has a flexible substrate.

23. For example, the ZTE Axon 30 Ultra smartphone includes a flexible curved AMOLED (Organic Light Emitting Diode) display, which includes a front panel and a back panel. The AMOLED display includes multiple layers. The top layer ("front panel") of the display includes a rigid structure: Corning Gorilla Glass 5 ("rigid substrate") as the cover glass.



Display 6.67 inches, FHD+ (2400×1080) 20:9 144Hz , AMOLED flexible curved screen

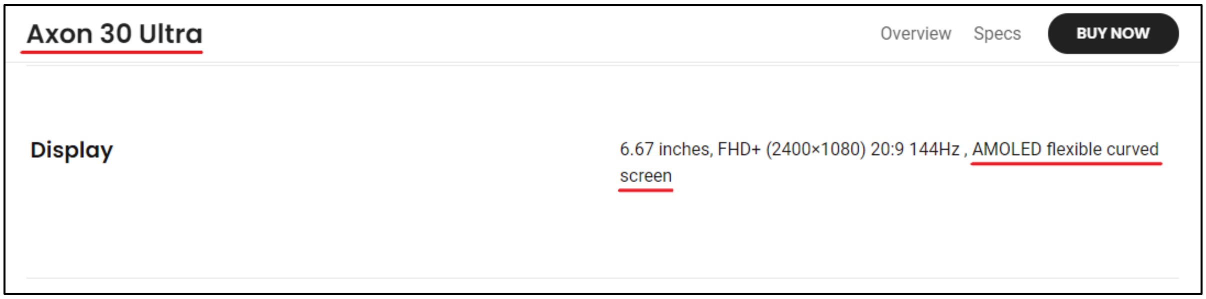
Source: https://ztedevices.com/en-us/axon-30-ultra-specs/



That big screen makes the Axon 30 Ultra a big device. It's comfortable enough to hold, and the curved screen and back panel help, but I definitely can't reach my thumb clear across the screen while using it one-handed. There's Gorilla Glass 5 on the front and back panels and the frame is aluminum, giving the whole phone a solid, premium feel.

Source: https://www.theverge.com/22458629/zte-axon-30-ultra-review-screen-price-specs

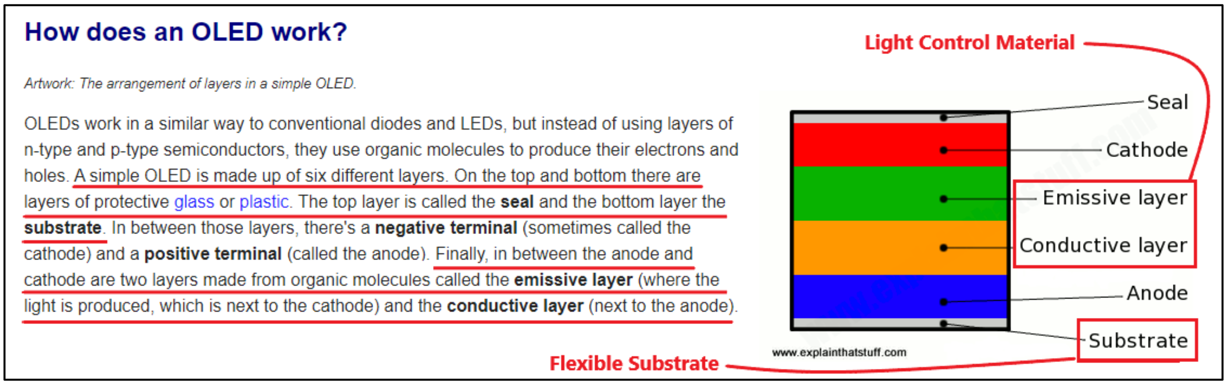
24. The display also includes an OLED material ("light control material") as the middle layer between the front panel and the back panel. This layer includes emissive and conductive layers. The emissive and conductive layers ("light control material") are responsible for emitting light and are made of organic molecules.



Axon 30 Ultra

Display	6.67 inches, FHD+ (2400×1080) 20:9 144Hz , AMOLED flexible curved screen

Source: https://ztedevices.com/en-us/axon-30-ultra-specs/

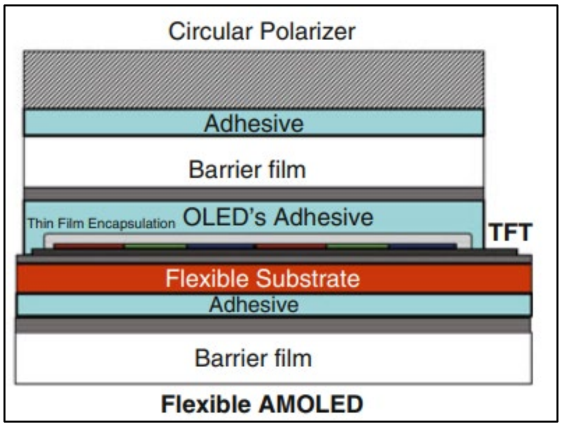


How does an OLED work?

Artwork: The arrangement of layers in a simple OLED.

OLEDs work in a similar way to conventional diodes and LEDs, but instead of using layers of n-type and p-type semiconductors, they use organic molecules to produce their electrons and holes. A simple OLED is made up of six different layers. On the top and bottom there are layers of protective glass or plastic. The top layer is called the **seal** and the bottom layer the **substrate**. In between those layers, there's a **negative terminal** (sometimes called the cathode) and a **positive terminal** (called the anode). Finally, in between the anode and cathode are two layers made from organic molecules called the **emissive layer** (where the light is produced, which is next to the cathode) and the **conductive layer** (next to the anode).

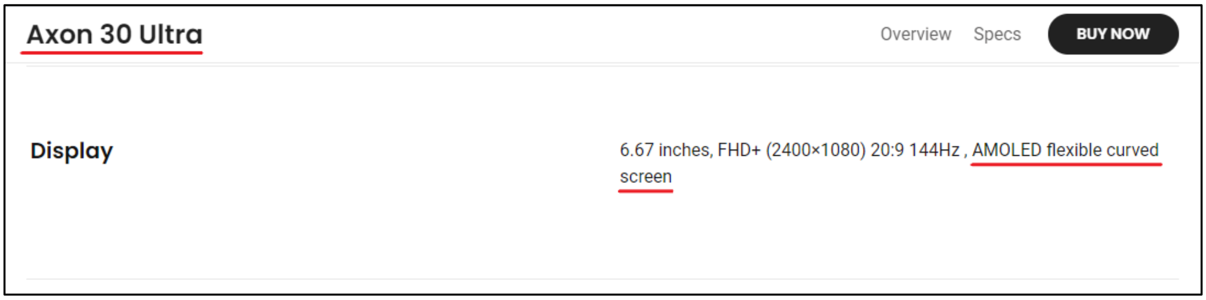
Source: https://www.explainthatstuff.com/how-oleds-and-leps-work.html



Source: https://media.springernature.com/lw785/springer-static/image/chp%3A10.1007%2F978-3-319-14346-0_213/MediaObjects/150079_2_En_213_Fig1_HTML.gif

25. The bottom layer ("back panel") is a flexible plastic substrate layer ("flexible substrate"). In particular, the bottom layer includes multiple organic materials, such as polymeric materials (e.g., flexible plastic films). Polymers, such as polyimide (PI), polyethylene terephthalate (PET), etc., are commonly used in the manufacture of flexible displays (and, in

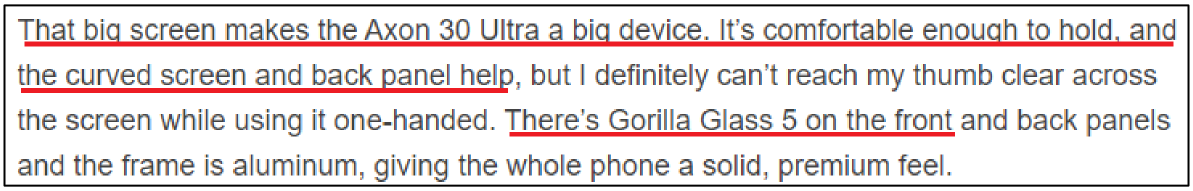
particular, flexible AMOLED displays). Additionally, flexible display materials are used to make curved screen or curved edge displays, like the ZTE Axon 30 Ultra display.



Axon 30 Ultra

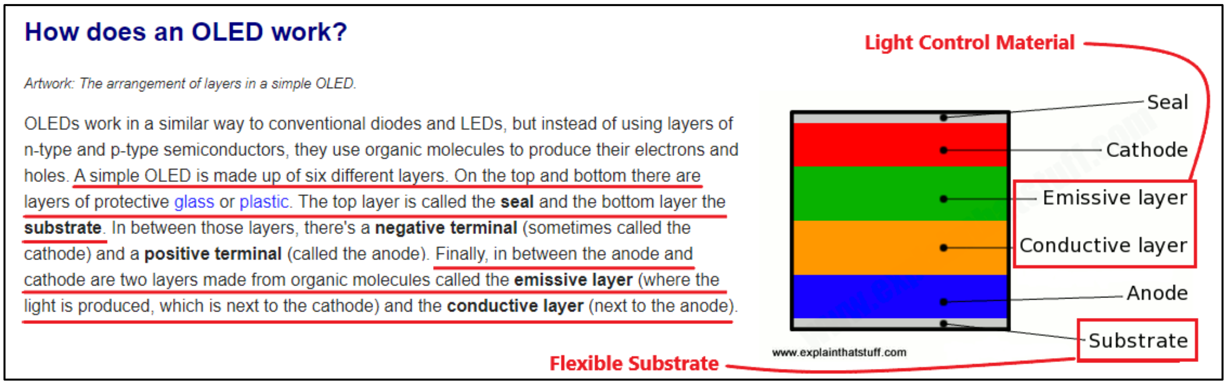
Display 6.67 inches, FHD+ (2400×1080) 20:9 144Hz , AMOLED flexible curved screen

Source: https://ztedevices.com/en-us/axon-30-ultra-specs/



That big screen makes the Axon 30 Ultra a big device. It's comfortable enough to hold, and the curved screen and back panel help, but I definitely can't reach my thumb clear across the screen while using it one-handed. There's Gorilla Glass 5 on the front and back panels and the frame is aluminum, giving the whole phone a solid, premium feel.

Source: https://www.theverge.com/22458629/zte-axon-30-ultra-review-screen-price-specs

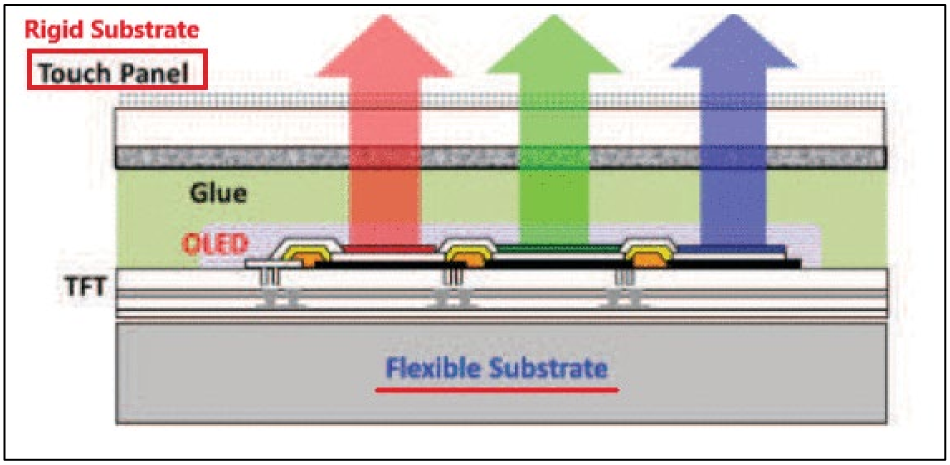


How does an OLED work?

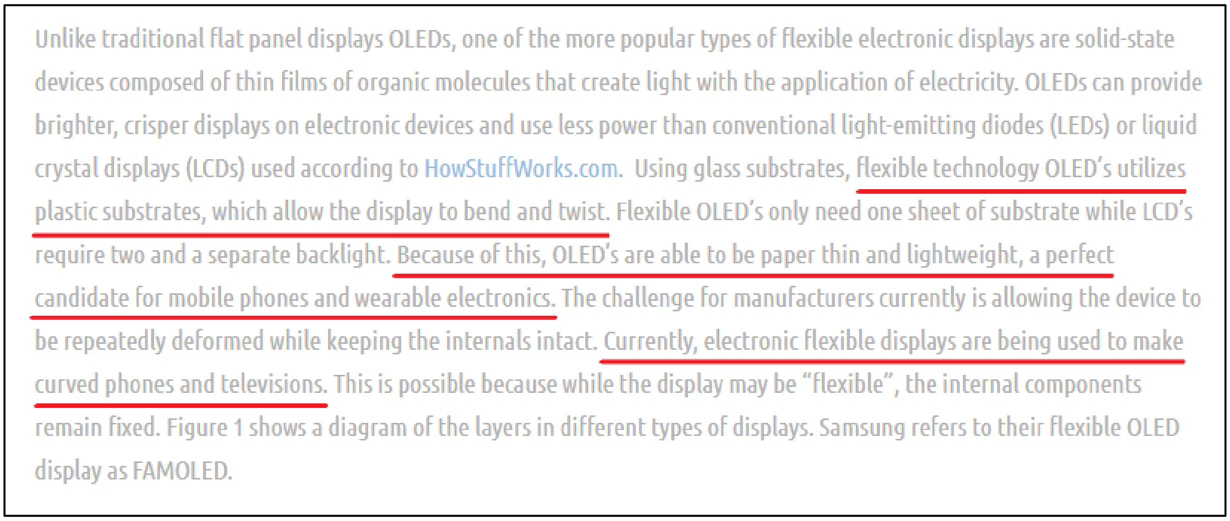
Artwork: The arrangement of layers in a simple OLED.

OLEDs work in a similar way to conventional diodes and LEDs, but instead of using layers of n-type and p-type semiconductors, they use organic molecules to produce their electrons and holes. A simple OLED is made up of six different layers. On the top and bottom there are layers of protective glass or plastic. The top layer is called the **seal** and the bottom layer the **substrate**. In between those layers, there's a **negative terminal** (sometimes called the cathode) and a **positive terminal** (called the anode). Finally, in between the anode and cathode are two layers made from organic molecules called the **emissive layer** (where the light is produced, which is next to the cathode) and the **conductive layer** (next to the anode).

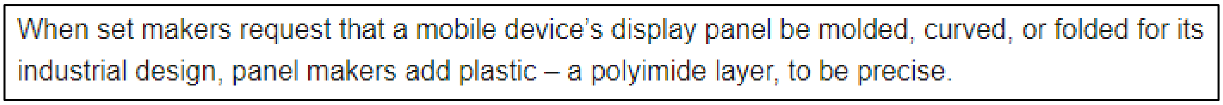
Source: https://www.explainthatstuff.com/how-oleds-and-leps-work.html



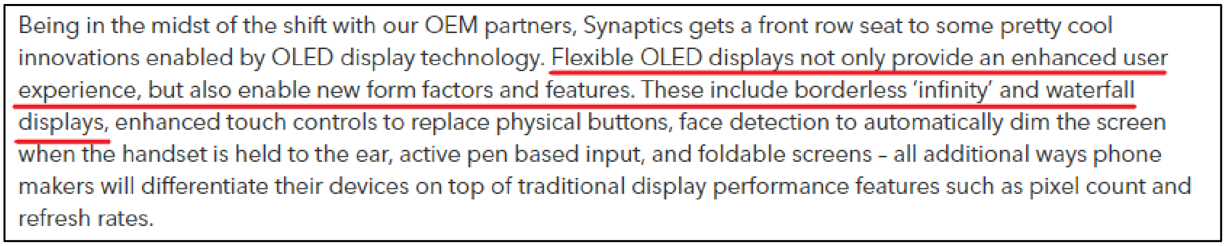
Source: https://ieeexplore.ieee.org/abstract/document/6514818


Unlike traditional flat panel displays OLEDs, one of the more popular types of flexible electronic displays are solid-state devices composed of thin films of organic molecules that create light with the application of electricity. OLEDs can provide brighter, crisper displays on electronic devices and use less power than conventional light-emitting diodes (LEDs) or liquid crystal displays (LCDs) used according to HowStuffWorks.com. Using glass substrates, flexible technology OLED's utilizes plastic substrates, which allow the display to bend and twist. Flexible OLED's only need one sheet of substrate while LCD's require two and a separate backlight. Because of this, OLED's are able to be paper thin and lightweight, a perfect candidate for mobile phones and wearable electronics. The challenge for manufacturers currently is allowing the device to be repeatedly deformed while keeping the internals intact. Currently, electronic flexible displays are being used to make curved phones and televisions. This is possible because while the display may be "flexible", the internal components remain fixed. Figure 1 shows a diagram of the layers in different types of displays. Samsung refers to their flexible OLED display as FAMOLED.

Source: https://patinformatics.com/are-flexible-electronic-displays-the-future-of-smartphone-display-technology-guest-post-by-riley-collins/

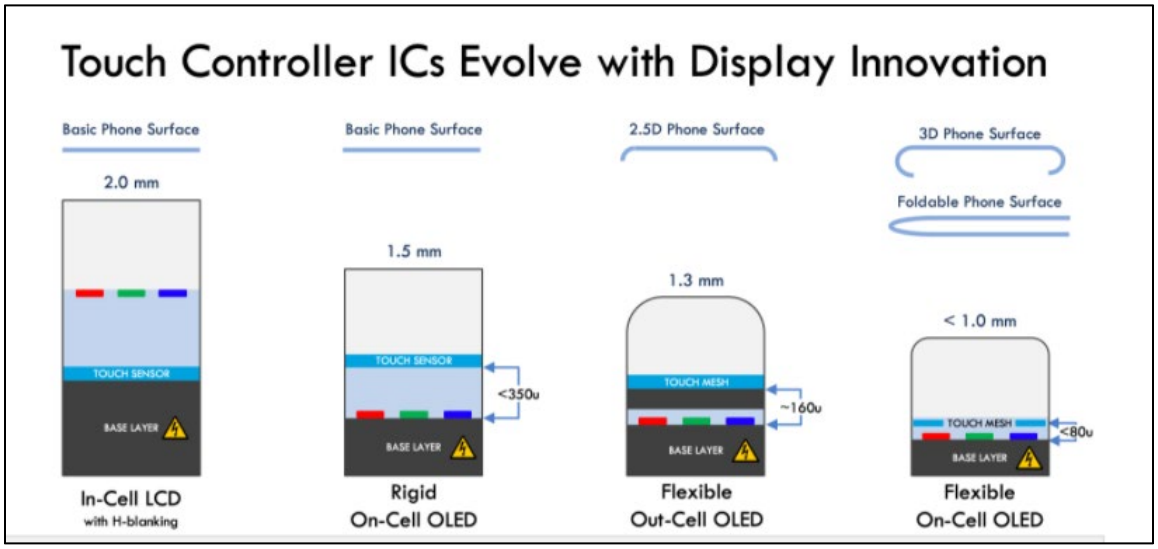

When set makers request that a mobile device's display panel be molded, curved, or folded for its industrial design, panel makers add plastic – a polyimide layer, to be precise.

Source: https://www.corning.com/worldwide/en/products/display-glass/carrying-handheld-devices-into-the-flexible-oled-future.html

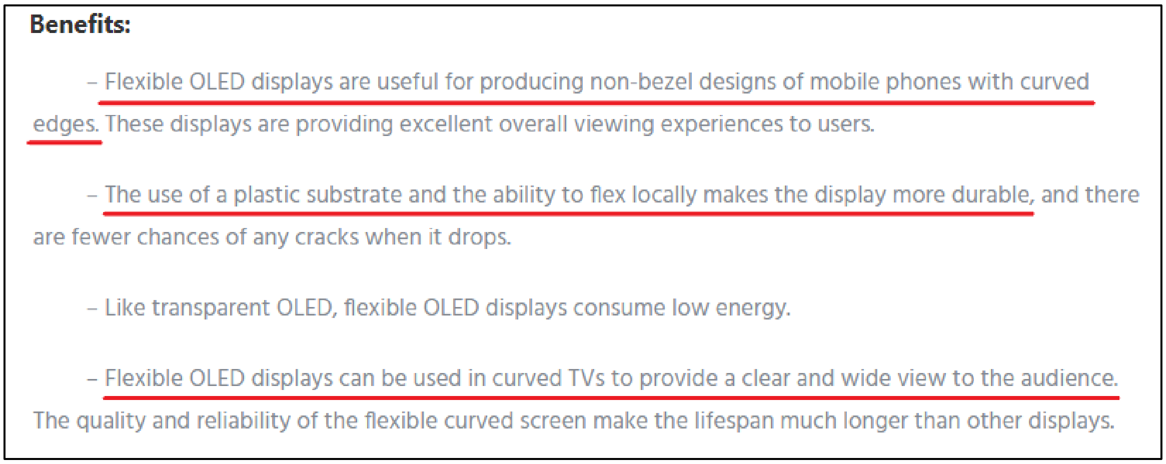


Being in the midst of the shift with our OEM partners, Synaptics gets a front row seat to some pretty cool innovations enabled by OLED display technology. Flexible OLED displays not only provide an enhanced user experience, but also enable new form factors and features. These include borderless 'infinity' and waterfall displays, enhanced touch controls to replace physical buttons, face detection to automatically dim the screen when the handset is held to the ear, active pen based input, and foldable screens – all additional ways phone makers will differentiate their devices on top of traditional display performance features such as pixel count and refresh rates.

Source: https://www.synaptics.com/company/blog/flexible-oled-on-cell-key



Source: https://www.synaptics.com/company/blog/flexible-oled-on-cell-key



Benefits:

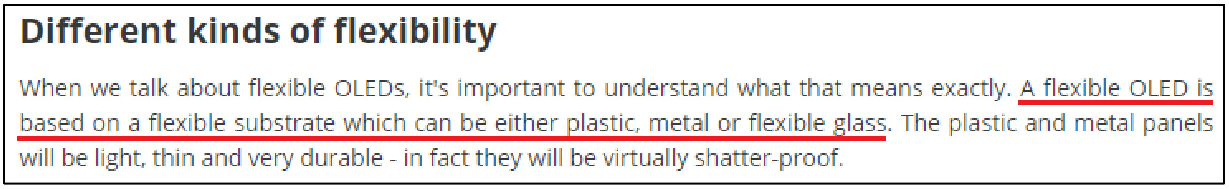
– Flexible OLED displays are useful for producing non-bezel designs of mobile phones with curved edges. These displays are providing excellent overall viewing experiences to users.

– The use of a plastic substrate and the ability to flex locally makes the display more durable, and there are fewer chances of any cracks when it drops.

– Like transparent OLED, flexible OLED displays consume low energy.

– Flexible OLED displays can be used in curved TVs to provide a clear and wide view to the audience. The quality and reliability of the flexible curved screen make the lifespan much longer than other displays.

Source: https://www.e3displays.com/transparent-and-flexible-oled-displays-are-transforming-traditional-viewing/



Different kinds of flexibility

When we talk about flexible OLEDs, it's important to understand what that means exactly. A flexible OLED is based on a flexible substrate which can be either plastic, metal or flexible glass. The plastic and metal panels will be light, thin and very durable - in fact they will be virtually shatter-proof.

Source: https://www.oled-info.com/flexible-oled

26. Accordingly, the front panel includes a rigid substrate, and the back panel includes a flexible substrate, with a light control material therebetween.

27. ZTE has had knowledge of the ’086 Patent at least as of the date when it was notified of the filing of this action, and as early as November 26, 2021, when ZTE received a letter from Buffalo notifying it of the ’086 Patent, that Buffalo Patents is the owner of the ’086 Patent, and that ZTE was infringing the ’086 Patent based on technology incorporated into ZTE smartphones, such as the ZTE Axon 30 Ultra smartphone.

28. Buffalo Patents has been damaged as a result of the infringing conduct by ZTE alleged above. Thus, ZTE is liable to Buffalo Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

29. Buffalo Patents has neither made nor sold unmarked articles that practice the ’086 Patent, and is entitled to collect pre-filing damages for the full period allowed by law for infringement of the ’086 Patent.

COUNT II

DIRECT INFRINGEMENT OF U.S. PATENT NO. 6,904,405

30. On June 7, 2005, the ’405 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Message Recognition Using Shared Language Model.”

31. Buffalo Patents is the owner of the ’405 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the ’405 Patent against infringers, and to collect damages for all relevant times.

32. ZTE made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, its ZTE Blade A7 Prime smartphone and other products[4] that include the Gboard application or similar virtual keyboard technology (“accused products”):

[4] *See, e.g.*, ZTE AVID (multiple models), ZTE Awe, ZTE AXON (multiple models), ZTE Blade (multiple models), ZTE Citrine LTE, ZTE Compel GoPhone, ZTE Cymbal-T, ZTE Fanfare, ZTE Fanfare 3, ZTE Flame, ZTE Fury, ZTE Quartz, ZTE Gabb (Z1, Z2), ZTE Grand (multiple models), ZTE Hawkeye, ZTE Imperial Max, ZTE JASPER, ZTE Libero 2, ZTE Majesty (Pro, Pro Plus), ZTE Maven (2, 3), ZTE Max (Blue LTE, Duo LTE, XL), ZTE NUBIA (multiple models), ZTE Obsidian, ZTE Overture 3, ZTE Prelude, ZTE Prelude 2, ZTE Prelude+, ZTE Prestige 2, ZTE Quest 5, ZTE S30, ZTE S30 Pro, ZTE S30 SE, ZTE small Fresh (4, 5, 5s), ZTE Sonata 3, ZTE Speed, ZTE Tempo, ZTE Tempo Go, ZTE Tempo X, ZTE Tough Max 2, ZTE V870, ZTE Visible R2, ZTE Vision R2, ZTE Voyage 4, ZTE Voyage 4S, ZTE Warp 7, ZTE Whirl 2, ZTE Z557, ZTE Z667 GoPhone, ZTE Zfive (multiple models), ZTE Zinger, ZTE ZMAX (multiple models), ZTE K88 Trek 2, ZTE K92 Primetime, ZTE ZPad (8”, 10”), ZTE V9, ZTE Gabb Smart Watch, ZTE Quartz, ZTE Spro, ZTE Spro 2, ZTE Spro Plus, etc.

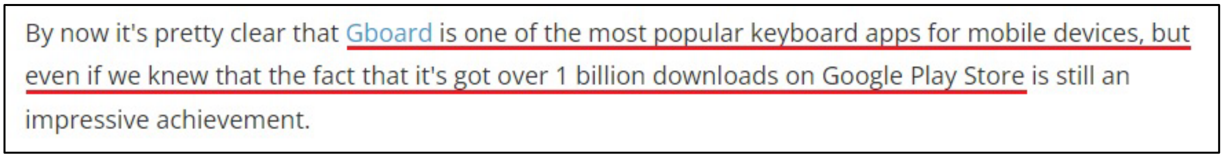
Source: https://zteusa.com/products/zte-blade-a7-prime

33. By doing so, ZTE has directly infringed (literally and/or under the doctrine of equivalents) at least Claim 7 of the '405 Patent. ZTE's infringement in this regard is ongoing.

34. The ZTE Blade A7 Prime smartphone is an exemplary accused product.

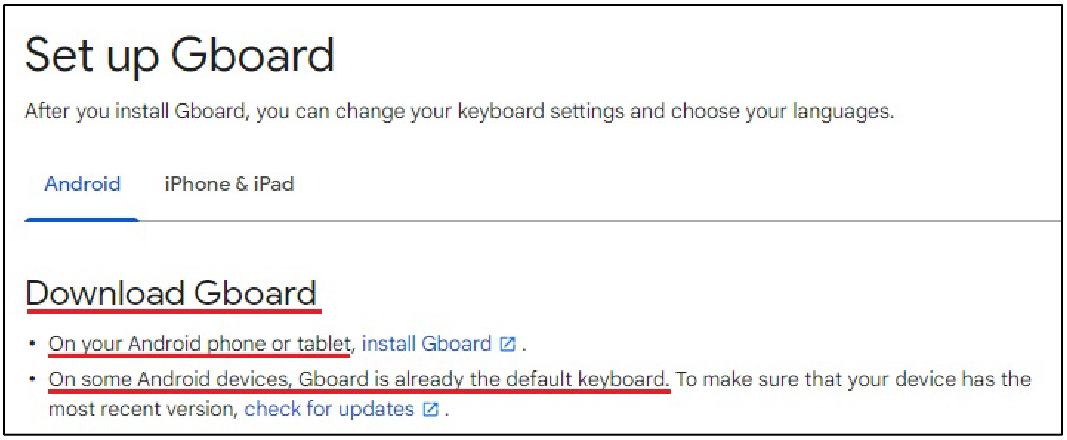
35. The ZTE Blade A7 Prime smartphone implements a method for performing message recognition with a shared language model.

36. For example, the ZTE Blade A7 Prime smartphone includes Gboard, which is the smartphone's default virtual keyboard application. The virtual keyboard application on the ZTE Blade A7 Prime smartphone includes different message recognition types, including voice to text and handwriting to text. The language model used by the smartphone is an n-gram language model ("shared language model").



By now it's pretty clear that Gboard is one of the most popular keyboard apps for mobile devices, but even if we knew that the fact that it's got over 1 billion downloads on Google Play Store is still an impressive achievement.

Source: https://www.phonearena.com/news/Google-Gboard-keyboard-app-1-billion-downloads_id108061



Set up Gboard

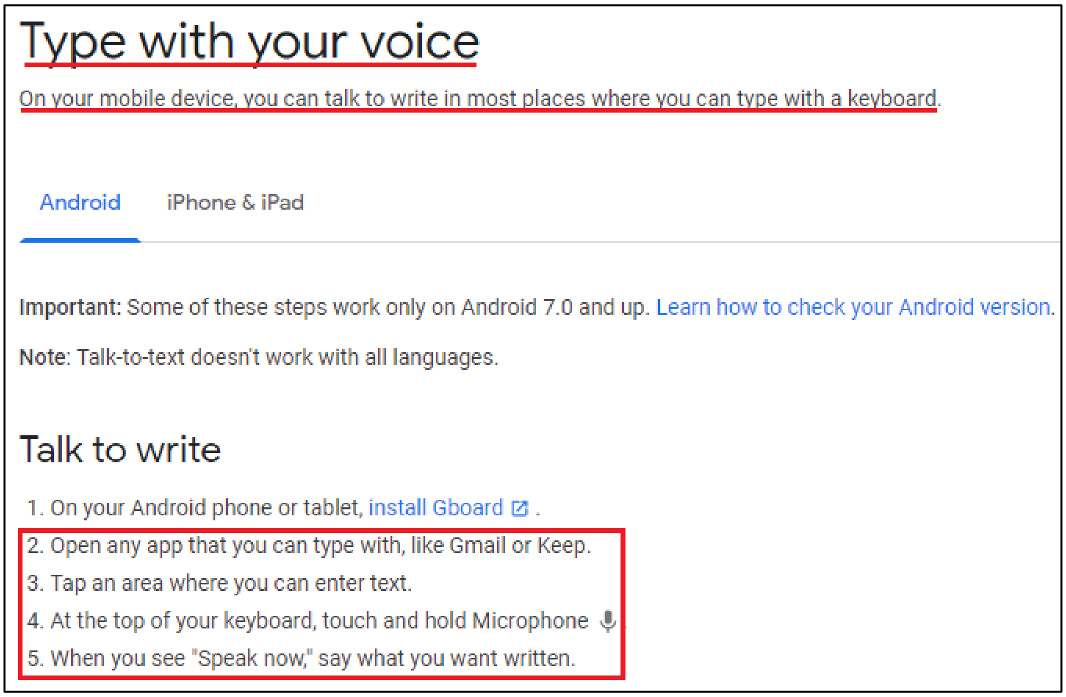
After you install Gboard, you can change your keyboard settings and choose your languages.

Download Gboard

- On your Android phone or tablet, install Gboard .
- On some Android devices, Gboard is already the default keyboard. To make sure that your device has the most recent version, check for updates .

Source:

https://support.google.com/gboard/answer/6380730?hl=en&co=GENIE.Platform%3DAndroid



Type with your voice

On your mobile device, you can talk to write in most places where you can type with a keyboard.

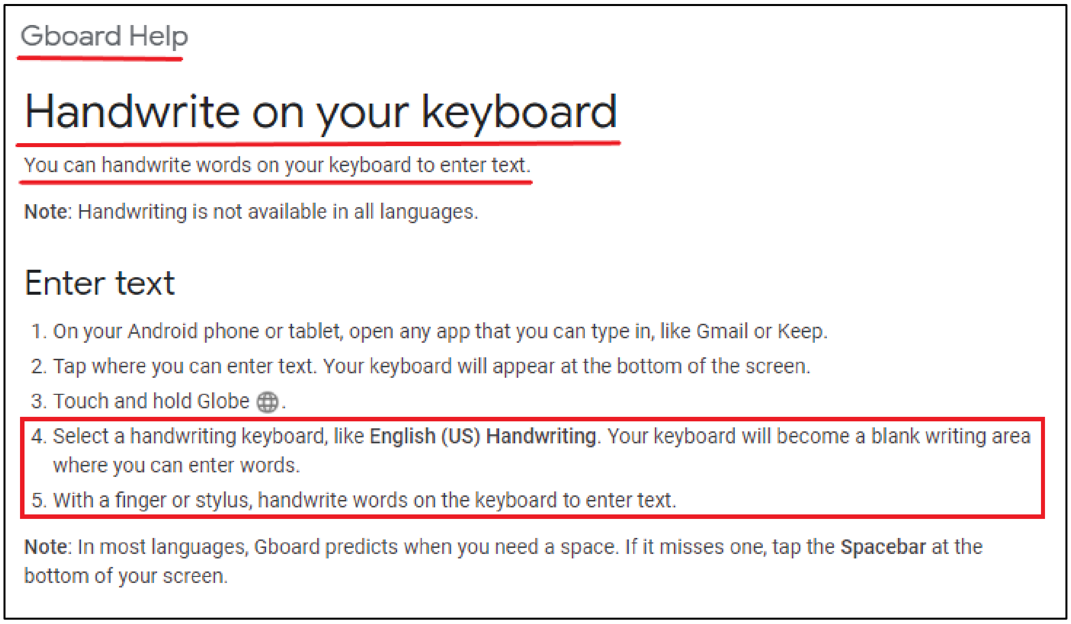
Important: Some of these steps work only on Android 7.0 and up. Learn how to check your Android version.

Note: Talk-to-text doesn't work with all languages.

Talk to write

1. On your Android phone or tablet, install Gboard .
2. Open any app that you can type with, like Gmail or Keep.
3. Tap an area where you can enter text.
4. At the top of your keyboard, touch and hold Microphone 🎤.
5. When you see "Speak now," say what you want written.

Source: https://support.google.com/gboard/answer/2781851?hl=en&ref_topic=9024098



Gboard Help

Handwrite on your keyboard

You can handwrite words on your keyboard to enter text.

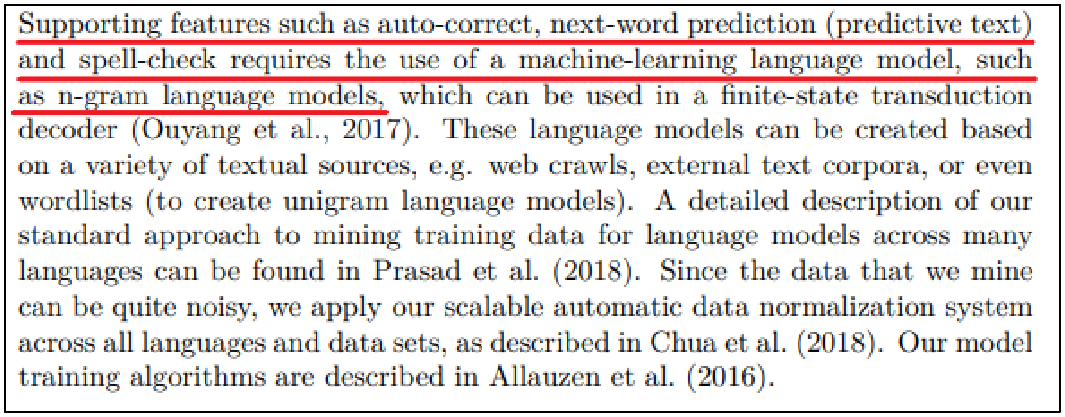
Note: Handwriting is not available in all languages.

Enter text

1. On your Android phone or tablet, open any app that you can type in, like Gmail or Keep.
2. Tap where you can enter text. Your keyboard will appear at the bottom of the screen.
3. Touch and hold Globe 🌐.
4. Select a handwriting keyboard, like **English (US) Handwriting**. Your keyboard will become a blank writing area where you can enter words.
5. With a finger or stylus, handwrite words on the keyboard to enter text.

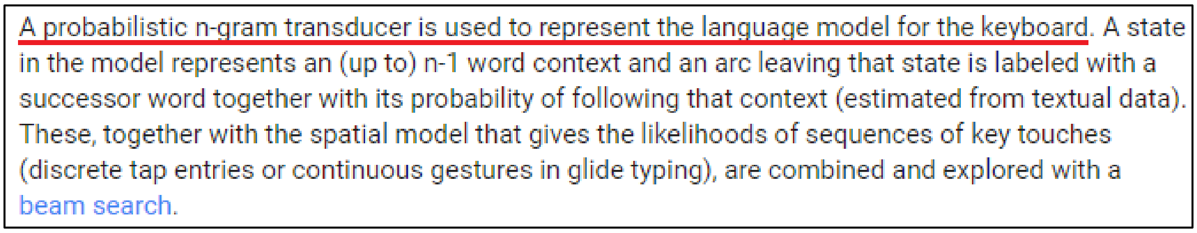
Note: In most languages, Gboard predicts when you need a space. If it misses one, tap the **Spacebar** at the bottom of your screen.

Source: https://support.google.com/gboard/answer/9108773?hl=en&ref_topic=9024098



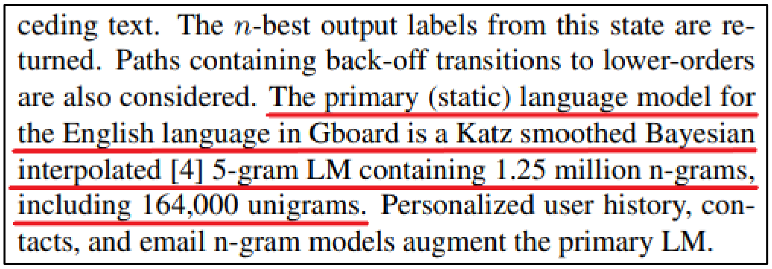
Supporting features such as auto-correct, next-word prediction (predictive text) and spell-check requires the use of a machine-learning language model, such as n-gram language models, which can be used in a finite-state transduction decoder (Ouyang et al., 2017). These language models can be created based on a variety of textual sources, e.g. web crawls, external text corpora, or even wordlists (to create unigram language models). A detailed description of our standard approach to mining training data for language models across many languages can be found in Prasad et al. (2018). Since the data that we mine can be quite noisy, we apply our scalable automatic data normalization system across all languages and data sets, as described in Chua et al. (2018). Our model training algorithms are described in Allauzen et al. (2016).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 16)



A probabilistic n-gram transducer is used to represent the language model for the keyboard. A state in the model represents an (up to) n-1 word context and an arc leaving that state is labeled with a successor word together with its probability of following that context (estimated from textual data). These, together with the spatial model that gives the likelihoods of sequences of key touches (discrete tap entries or continuous gestures in glide typing), are combined and explored with a beam search.

Source: https://ai.googleblog.com/2017/05/the-machine-intelligence-behind-gboard.html

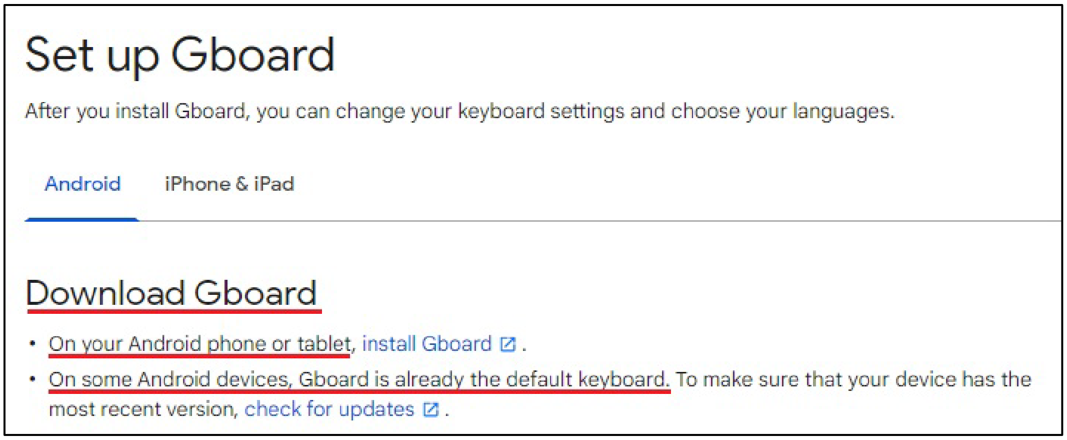


ceding text. The n-best output labels from this state are returned. Paths containing back-off transitions to lower-orders are also considered. The primary (static) language model for the English language in Gboard is a Katz smoothed Bayesian interpolated [4] 5-gram LM containing 1.25 million n-grams, including 164,000 unigrams. Personalized user history, contacts, and email n-gram models augment the primary LM.

Source: https://arxiv.org/pdf/1811.03604.pdf (Page 1)

37. The method implemented by the ZTE Blade A7 Prime smartphone includes performing message recognition of a first type, responsive to a first type of message input, to provide text data in accordance with both the shared language model and a first model specific to the first type of message recognition.

38. For example, the ZTE Blade A7 Prime smartphone includes Gboard, which is the smartphone's default virtual keyboard application. The virtual keyboard application on the ZTE Blade A7 Prime smartphone allows users to type a message with voice by converting the voice input into text. It uses an on-device, all neural speech recognizer. The speech recognizer ("message recognition of a first type") uses an end-to-end Recurrent Neural Network Transducer (RNN-T) model that combines an acoustic model, a pronunciation model, and a language model. The language model used by the smartphone is an n-gram language model. The user's voice input ("first type of message input") is converted into text using the speech recognizer. The pronunciation model, the acoustic model ("first model"), and the language model ("shared language model") can generate text responsive to voice input.



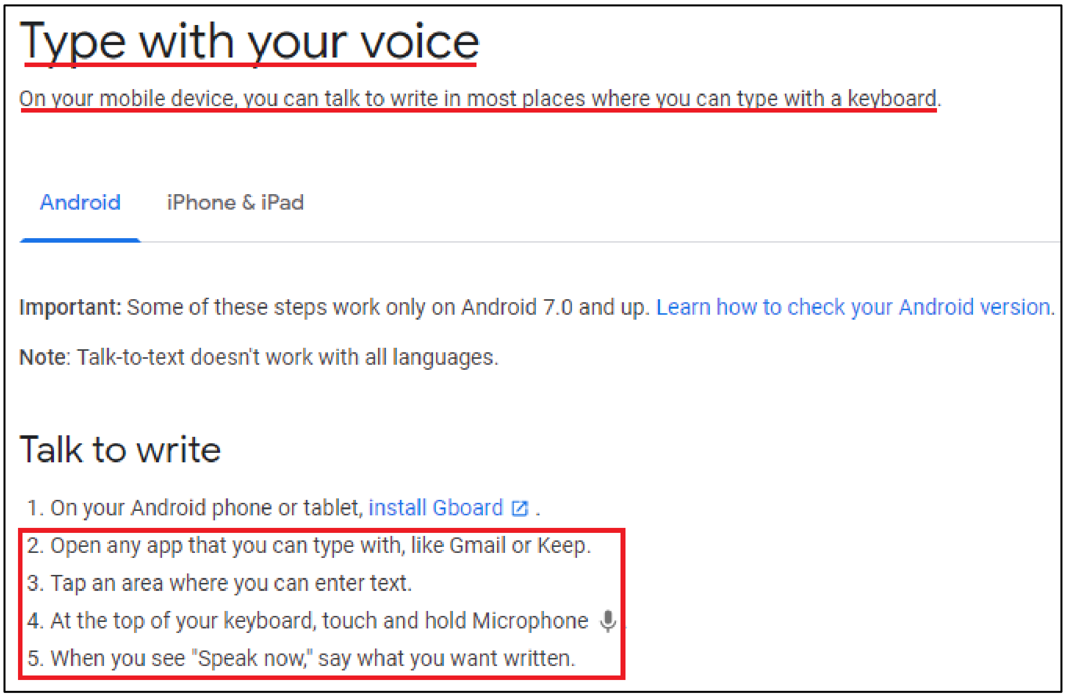
Set up Gboard

After you install Gboard, you can change your keyboard settings and choose your languages.

Download Gboard

- On your Android phone or tablet, install Gboard.
- On some Android devices, Gboard is already the default keyboard. To make sure that your device has the most recent version, check for updates.

Source:
https://support.google.com/gboard/answer/6380730?hl=en&co=GENIE.Platform%3DAndroid



Type with your voice

On your mobile device, you can talk to write in most places where you can type with a keyboard.

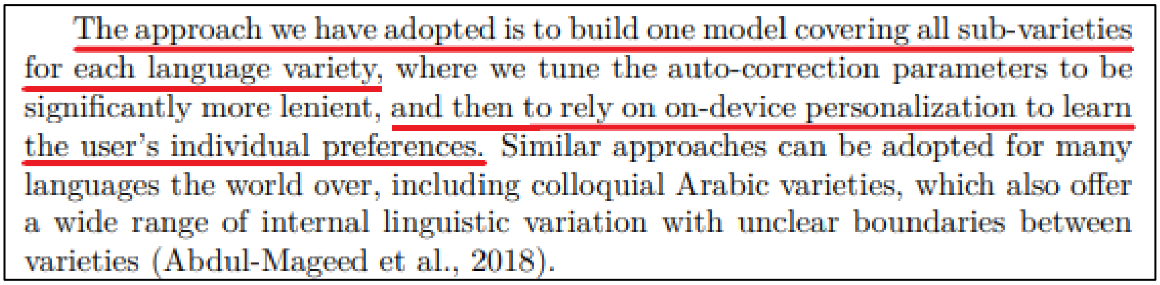
Important: Some of these steps work only on Android 7.0 and up. Learn how to check your Android version.

Note: Talk-to-text doesn't work with all languages.

Talk to write

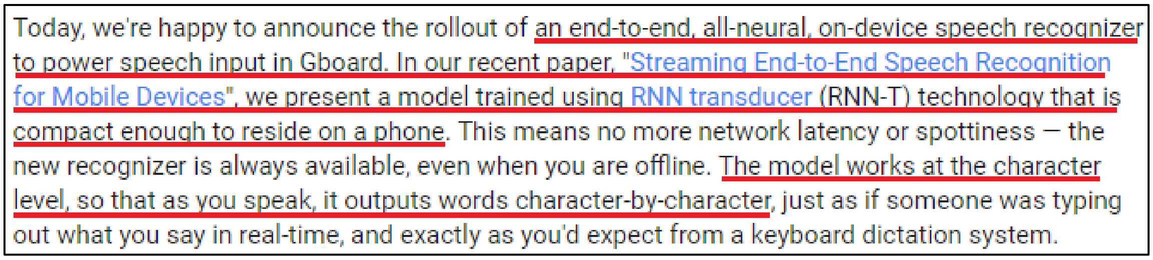
1. On your Android phone or tablet, install Gboard.
2. Open any app that you can type with, like Gmail or Keep.
3. Tap an area where you can enter text.
4. At the top of your keyboard, touch and hold Microphone 🎤.
5. When you see "Speak now," say what you want written.

Source: https://support.google.com/gboard/answer/2781851?hl=en&ref_topic=9024098



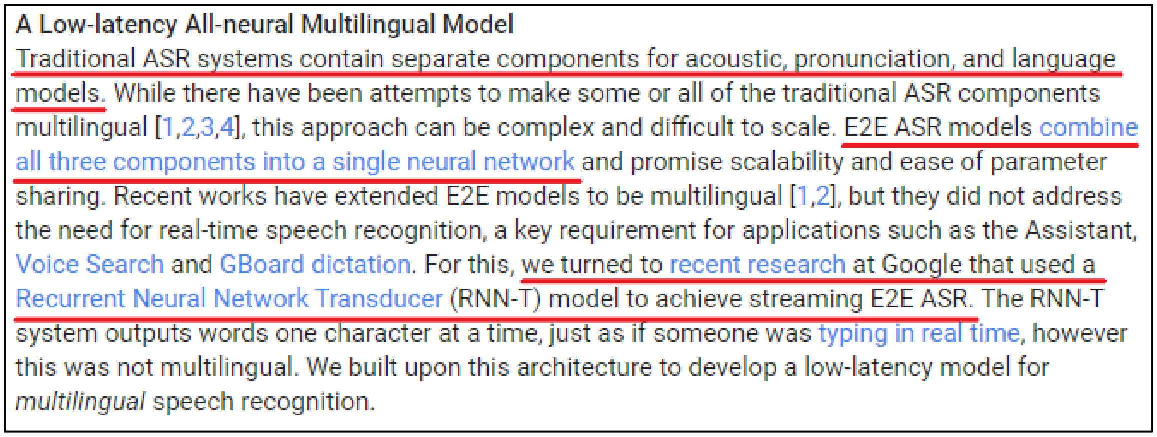
The approach we have adopted is to build one model covering all sub-varieties for each language variety, where we tune the auto-correction parameters to be significantly more lenient, and then to rely on on-device personalization to learn the user's individual preferences. Similar approaches can be adopted for many languages the world over, including colloquial Arabic varieties, which also offer a wide range of internal linguistic variation with unclear boundaries between varieties (Abdul-Mageed et al., 2018).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 17)



Today, we're happy to announce the rollout of an end-to-end, all-neural, on-device speech recognizer to power speech input in Gboard. In our recent paper, "Streaming End-to-End Speech Recognition for Mobile Devices", we present a model trained using RNN transducer (RNN-T) technology that is compact enough to reside on a phone. This means no more network latency or spottiness — the new recognizer is always available, even when you are offline. The model works at the character level, so that as you speak, it outputs words character-by-character, just as if someone was typing out what you say in real-time, and exactly as you'd expect from a keyboard dictation system.

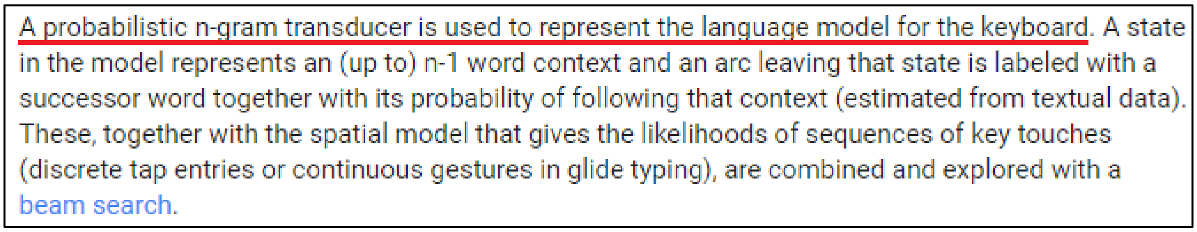
Source: https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html



A Low-latency All-neural Multilingual Model

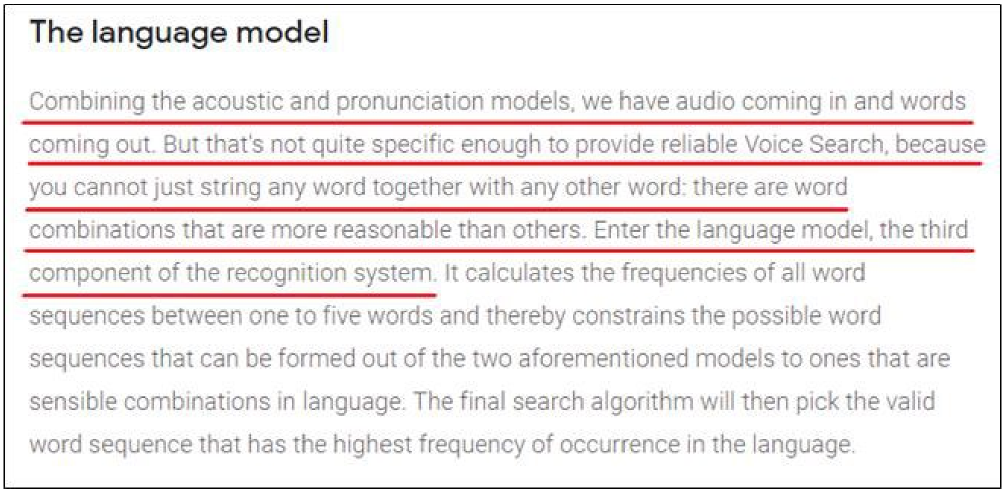
Traditional ASR systems contain separate components for acoustic, pronunciation, and language models. While there have been attempts to make some or all of the traditional ASR components multilingual [1,2,3,4], this approach can be complex and difficult to scale. E2E ASR models combine all three components into a single neural network and promise scalability and ease of parameter sharing. Recent works have extended E2E models to be multilingual [1,2], but they did not address the need for real-time speech recognition, a key requirement for applications such as the Assistant, Voice Search and GBoard dictation. For this, we turned to recent research at Google that used a Recurrent Neural Network Transducer (RNN-T) model to achieve streaming E2E ASR. The RNN-T system outputs words one character at a time, just as if someone was typing in real time, however this was not multilingual. We built upon this architecture to develop a low-latency model for *multilingual* speech recognition.

Source: https://ai.googleblog.com/2019/09/large-scale-multilingual-speech.html



A probabilistic n-gram transducer is used to represent the language model for the keyboard. A state in the model represents an (up to) n-1 word context and an arc leaving that state is labeled with a successor word together with its probability of following that context (estimated from textual data). These, together with the spatial model that gives the likelihoods of sequences of key touches (discrete tap entries or continuous gestures in glide typing), are combined and explored with a beam search.

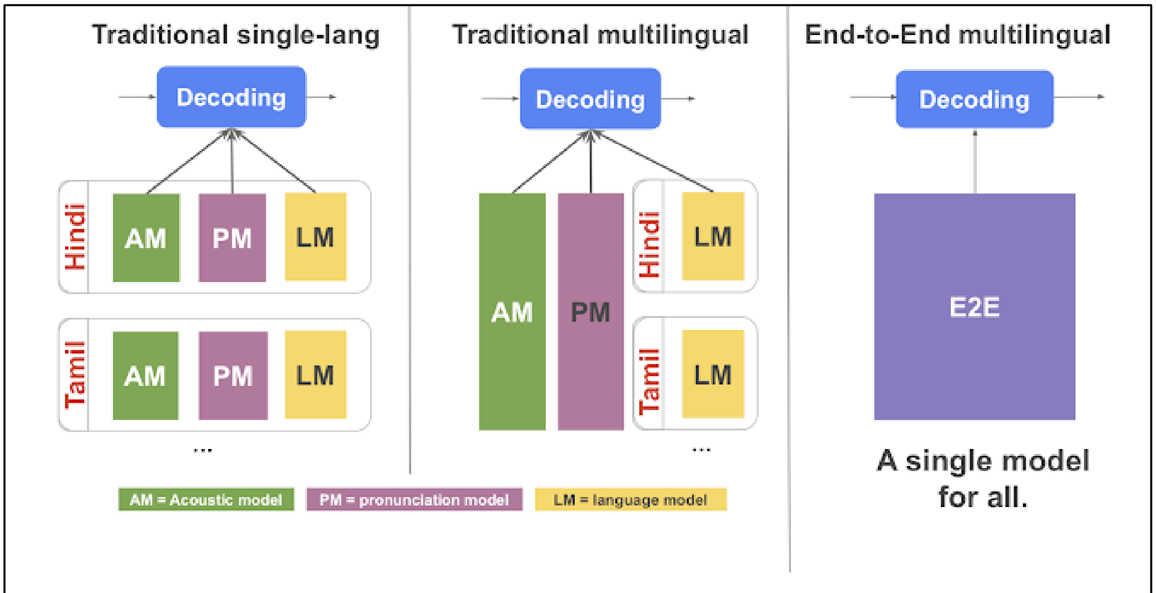
Source: https://ai.googleblog.com/2017/05/the-machine-intelligence-behind-gboard.html



The language model

Combining the acoustic and pronunciation models, we have audio coming in and words coming out. But that's not quite specific enough to provide reliable Voice Search, because you cannot just string any word together with any other word: there are word combinations that are more reasonable than others. Enter the language model, the third component of the recognition system. It calculates the frequencies of all word sequences between one to five words and thereby constrains the possible word sequences that can be formed out of the two aforementioned models to ones that are sensible combinations in language. The final search algorithm will then pick the valid word sequence that has the highest frequency of occurrence in the language.

Source: https://careers.google.com/stories/how-one-team-turned-the-dream-of-speech-recognition-into-a-reality/



[Left] A traditional monolingual speech recognizer comprising of Acoustic, Pronunciation and Language Models for each anguage. [Middle] A traditional multilingual speech recognizer where the Acoustic and Pronunciation model is multilingual, while the Language model is language-specific. [Right] An E2E multilingual speech recognizer where the Acoustic, Pronunciation and Language Model is combined into a single multilingual model.

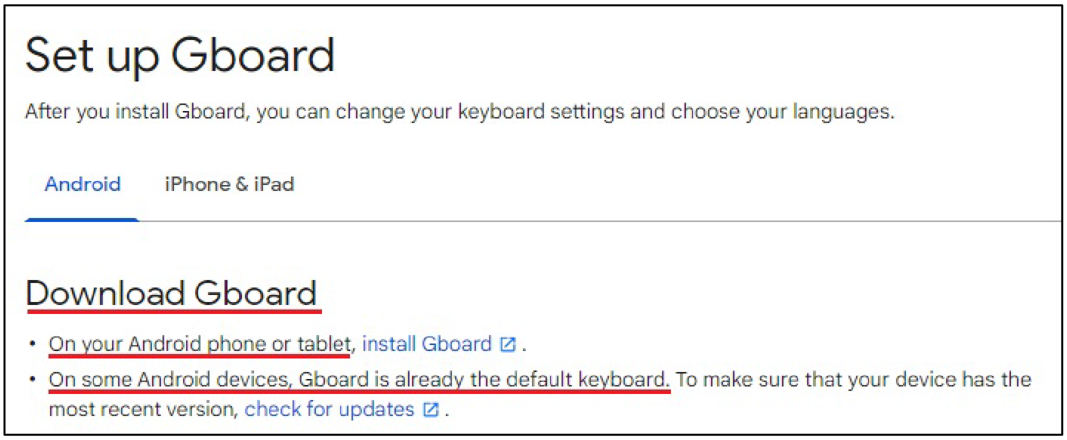
Source: https://ai.googleblog.com/2019/09/large-scale-multilingual-speech.html

39. The method implemented by the ZTE Blade A7 Prime smartphone includes performing message recognition of a second type, responsive to a second type of message input,

to provide text data in accordance with both the shared language model and a second model specific to the second type of message recognition.

40. For example, the ZTE Blade A7 Prime smartphone includes Gboard, which is the smartphone's default virtual keyboard application. The virtual keyboard application on the ZTE Blade A7 Prime smartphone provides the capability to convert a user's handwriting into text. It uses an end-to-end Recurrent Neural Network (RNN) model for handwriting recognition ("message recognition of a second type"). The language model used by the smartphone is an n-gram language model. The writing area on the device's keypad records the patterns created by the user, and the corresponding handwriting information ("second type of message input") is converted into text.

41. The handwriting model ("second model") used by the smartphone is a bi-directional version of the quasi-recurrent neural network (QRNN) model. The handwriting model, in combination with the n-gram language model ("shared language model"), can generate text responsive to handwriting input.



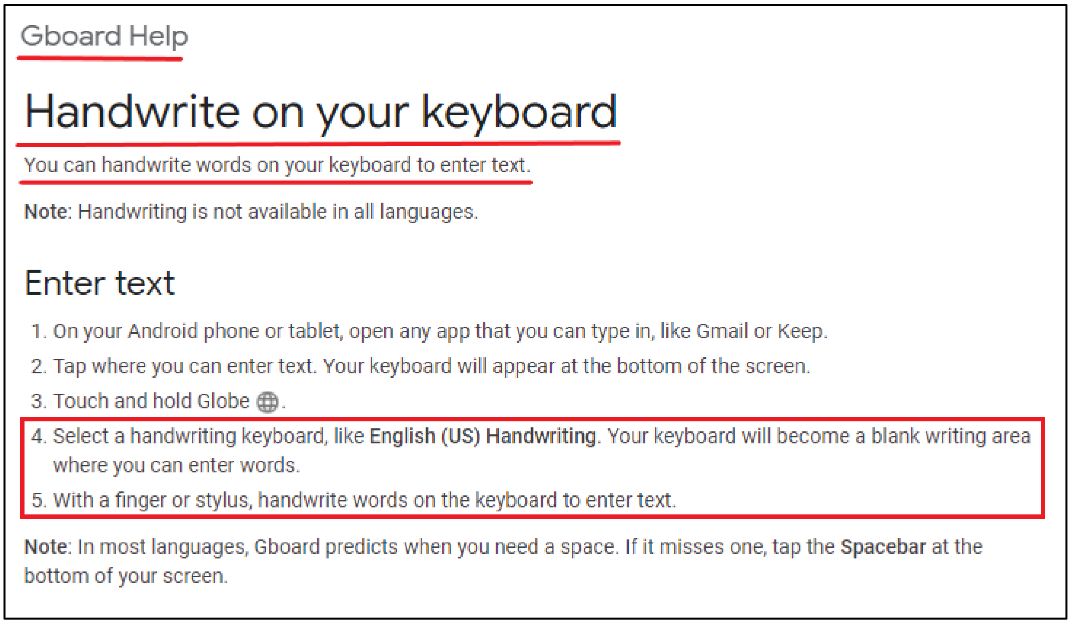
Set up Gboard

After you install Gboard, you can change your keyboard settings and choose your languages.

Download Gboard

- On your Android phone or tablet, install Gboard .
- On some Android devices, Gboard is already the default keyboard. To make sure that your device has the most recent version, check for updates .

Source:
https://support.google.com/gboard/answer/6380730?hl=en&co=GENIE.Platform%3DAndroid



Gboard Help

Handwrite on your keyboard

You can handwrite words on your keyboard to enter text.

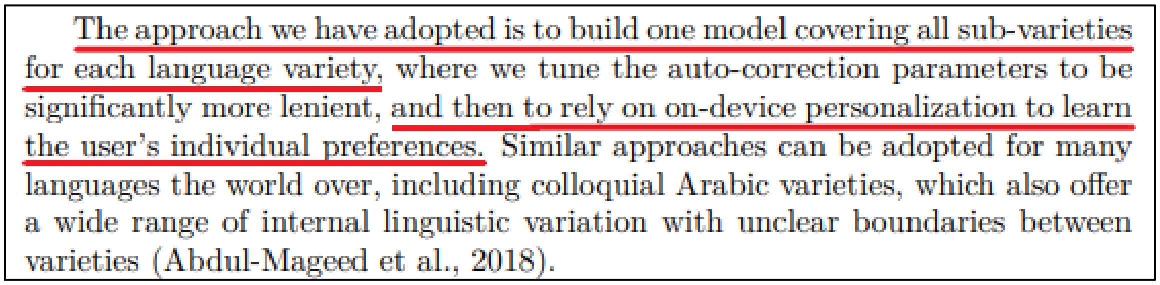
Note: Handwriting is not available in all languages.

Enter text

1. On your Android phone or tablet, open any app that you can type in, like Gmail or Keep.
2. Tap where you can enter text. Your keyboard will appear at the bottom of the screen.
3. Touch and hold Globe 🌐.
4. Select a handwriting keyboard, like **English (US) Handwriting**. Your keyboard will become a blank writing area where you can enter words.
5. With a finger or stylus, handwrite words on the keyboard to enter text.

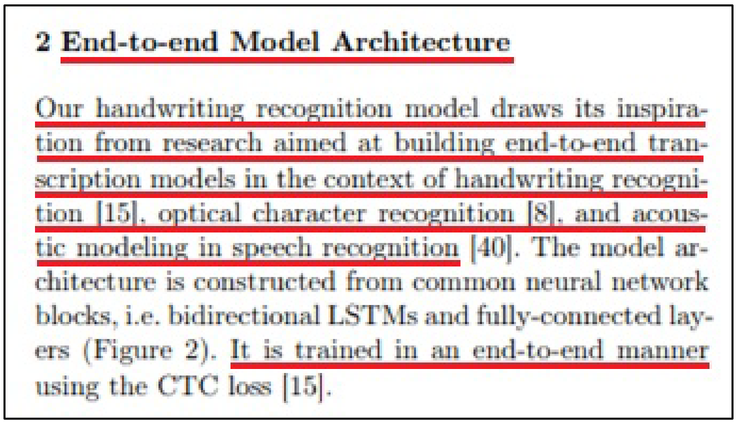
Note: In most languages, Gboard predicts when you need a space. If it misses one, tap the **Spacebar** at the bottom of your screen.

Source: https://support.google.com/gboard/answer/9108773?hl=en&ref_topic=9024098



The approach we have adopted is to build one model covering all sub-varieties for each language variety, where we tune the auto-correction parameters to be significantly more lenient, and then to rely on on-device personalization to learn the user's individual preferences. Similar approaches can be adopted for many languages the world over, including colloquial Arabic varieties, which also offer a wide range of internal linguistic variation with unclear boundaries between varieties (Abdul-Mageed et al., 2018).

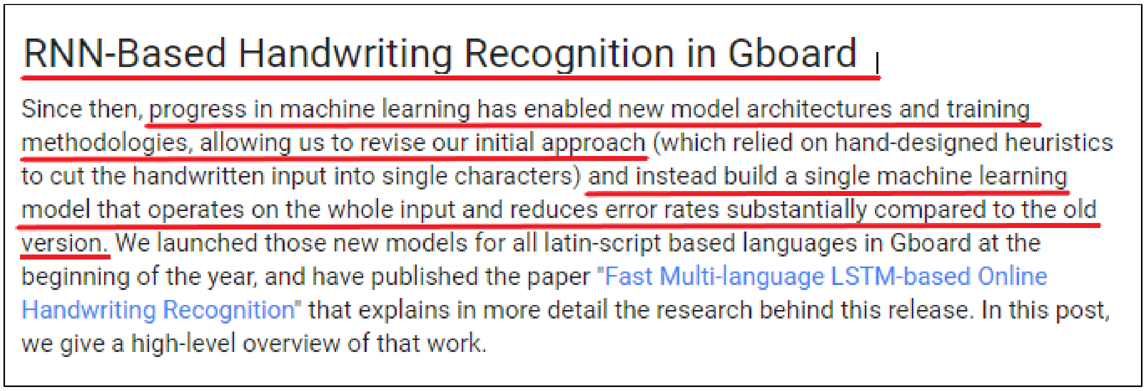
Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 17)



2 End-to-end Model Architecture

Our handwriting recognition model draws its inspiration from research aimed at building end-to-end transcription models in the context of handwriting recognition [15], optical character recognition [8], and acoustic modeling in speech recognition [40]. The model architecture is constructed from common neural network blocks, i.e. bidirectional LSTMs and fully-connected layers (Figure 2). It is trained in an end-to-end manner using the CTC loss [15].

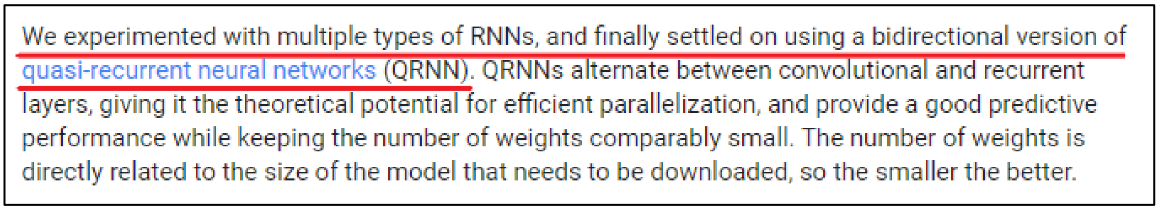
Source: https://arxiv.org/pdf/1902.10525.pdf (Page 3)



RNN-Based Handwriting Recognition in Gboard

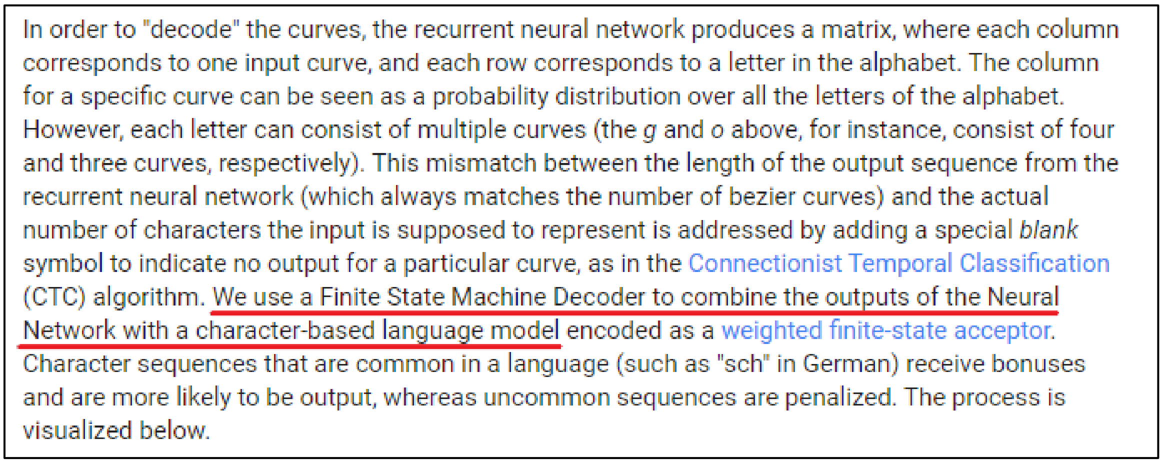
Since then, progress in machine learning has enabled new model architectures and training methodologies, allowing us to revise our initial approach (which relied on hand-designed heuristics to cut the handwritten input into single characters) and instead build a single machine learning model that operates on the whole input and reduces error rates substantially compared to the old version. We launched those new models for all latin-script based languages in Gboard at the beginning of the year, and have published the paper "Fast Multi-language LSTM-based Online Handwriting Recognition" that explains in more detail the research behind this release. In this post, we give a high-level overview of that work.

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html



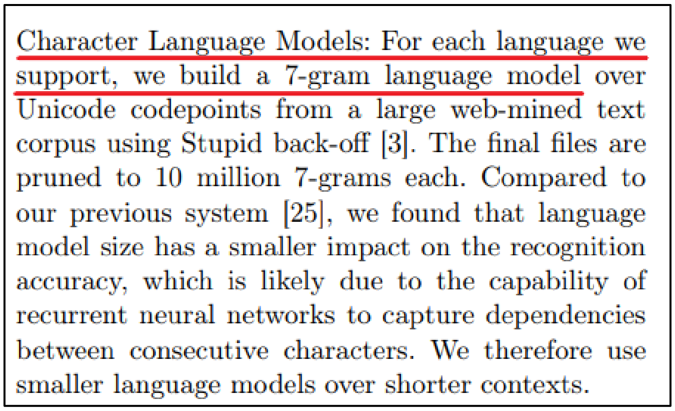
We experimented with multiple types of RNNs, and finally settled on using a bidirectional version of quasi-recurrent neural networks (QRNN). QRNNs alternate between convolutional and recurrent layers, giving it the theoretical potential for efficient parallelization, and provide a good predictive performance while keeping the number of weights comparably small. The number of weights is directly related to the size of the model that needs to be downloaded, so the smaller the better.

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html



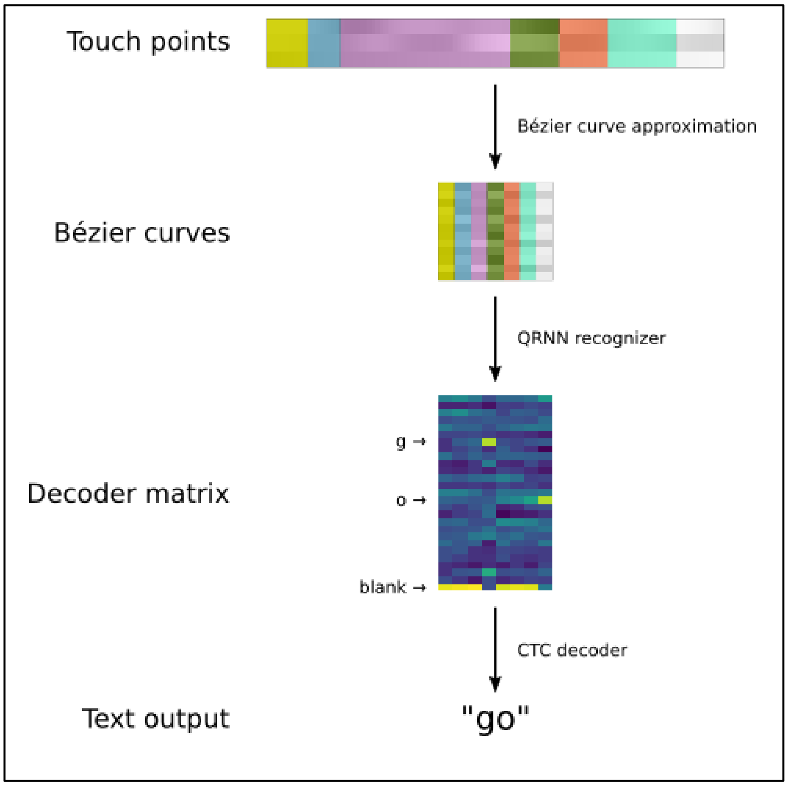
In order to "decode" the curves, the recurrent neural network produces a matrix, where each column corresponds to one input curve, and each row corresponds to a letter in the alphabet. The column for a specific curve can be seen as a probability distribution over all the letters of the alphabet. However, each letter can consist of multiple curves (the *g* and *o* above, for instance, consist of four and three curves, respectively). This mismatch between the length of the output sequence from the recurrent neural network (which always matches the number of bezier curves) and the actual number of characters the input is supposed to represent is addressed by adding a special *blank* symbol to indicate no output for a particular curve, as in the Connectionist Temporal Classification (CTC) algorithm. We use a Finite State Machine Decoder to combine the outputs of the Neural Network with a character-based language model encoded as a weighted finite-state acceptor. Character sequences that are common in a language (such as "sch" in German) receive bonuses and are more likely to be output, whereas uncommon sequences are penalized. The process is visualized below.

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html



Character Language Models: For each language we support, we build a 7-gram language model over Unicode codepoints from a large web-mined text corpus using Stupid back-off [3]. The final files are pruned to 10 million 7-grams each. Compared to our previous system [25], we found that language model size has a smaller impact on the recognition accuracy, which is likely due to the capability of recurrent neural networks to capture dependencies between consecutive characters. We therefore use smaller language models over shorter contexts.

Source: https://arxiv.org/pdf/1902.10525.pdf



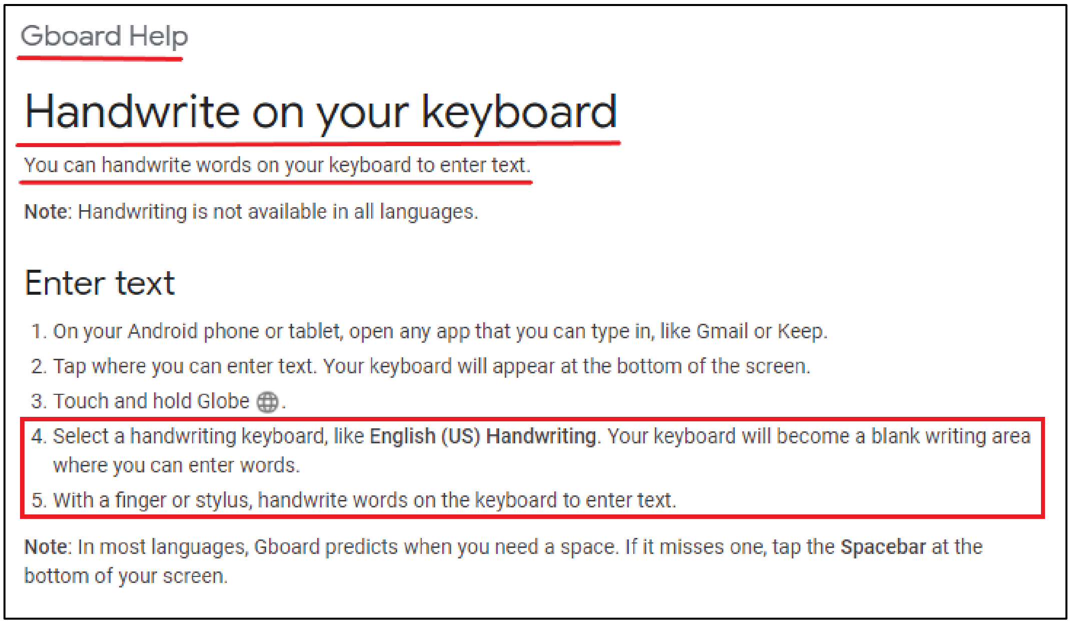
Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html

42. The method implemented by the ZTE Blade A7 Prime smartphone includes training the shared language model responsive to user correction of error in message recognition of either of the first and second types, thereby improving accuracy of each of the first and second types of message recognition.

43. For example, the ZTE Blade A7 Prime smartphone includes Gboard, which is the smartphone's default virtual keyboard application. The virtual keyboard application on the ZTE Blade A7 Prime smartphone provides the capability to convert a user's voice and/or handwriting into text. It provides users with a personalized model to learn the user's preferences. In the writing area of the keypad of the device, patterns created by the user are recorded, and the corresponding handwriting information is converted into text. The data input on the virtual keyboard is used to learn the user's preferences, train the language model to suit the user, and update the model locally. The smartphone builds one model for each language variety and uses on-device personalization to improve the language model by learning the user's preferences. Using an end-to-end model approach improves the predictions of both the speech and handwriting recognizers ("improving accuracy of each of the first and second types of message recognition").

44. As one example, when a user handwrites text in the writing area of the virtual keyboard, multiple suggestions can be provided to the user on the display area or interface (e.g., three suggestions to the left, center, and right). If the user selects the suggested text on the left or the right ("user correction") for the same written text multiple times, the user's selection is used to train the language model. If the user handwrites the same text at a later time, the user's previous selection is outputted as the primary suggestion (e.g., the suggested text in the center of the display area or interface).

45. Accordingly, the shared language model can be trained to be responsive to user correction of error in handwriting recognition.



Gboard Help

Handwrite on your keyboard

You can handwrite words on your keyboard to enter text.

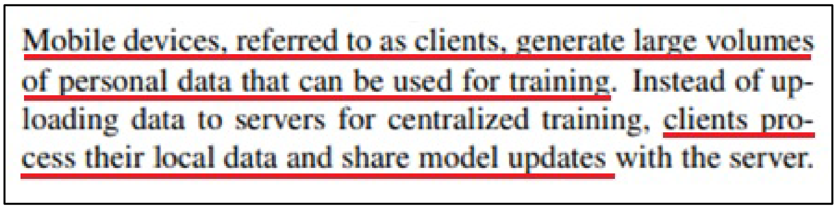
Note: Handwriting is not available in all languages.

Enter text

1. On your Android phone or tablet, open any app that you can type in, like Gmail or Keep.
2. Tap where you can enter text. Your keyboard will appear at the bottom of the screen.
3. Touch and hold Globe 🌐.
4. Select a handwriting keyboard, like **English (US) Handwriting**. Your keyboard will become a blank writing area where you can enter words.
5. With a finger or stylus, handwrite words on the keyboard to enter text.

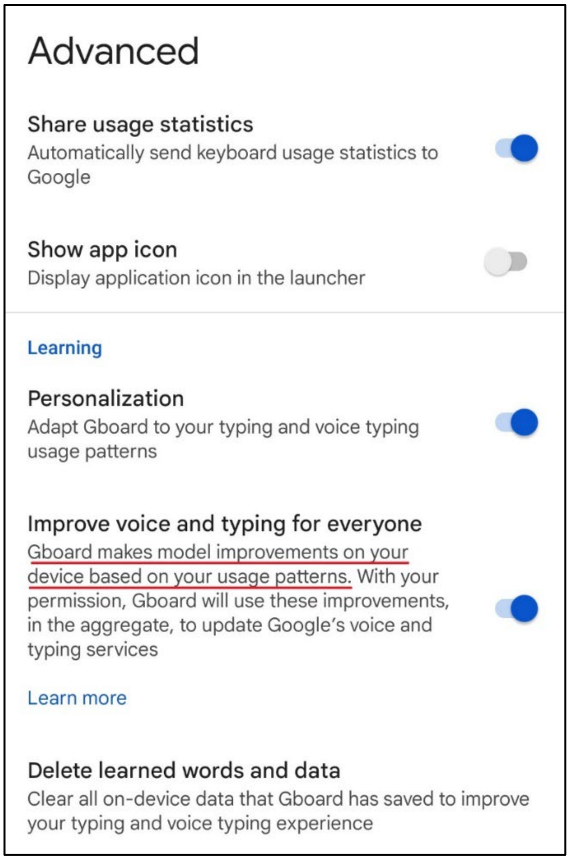
Note: In most languages, Gboard predicts when you need a space. If it misses one, tap the **Spacebar** at the bottom of your screen.

Source: https://support.google.com/gboard/answer/9108773?hl=en&ref_topic=9024098



Mobile devices, referred to as clients, generate large volumes of personal data that can be used for training. Instead of uploading data to servers for centralized training, clients process their local data and share model updates with the server.

Source: https://arxiv.org/pdf/1811.03604.pdf



Advanced

Share usage statistics
Automatically send keyboard usage statistics to Google

Show app icon
Display application icon in the launcher

Learning

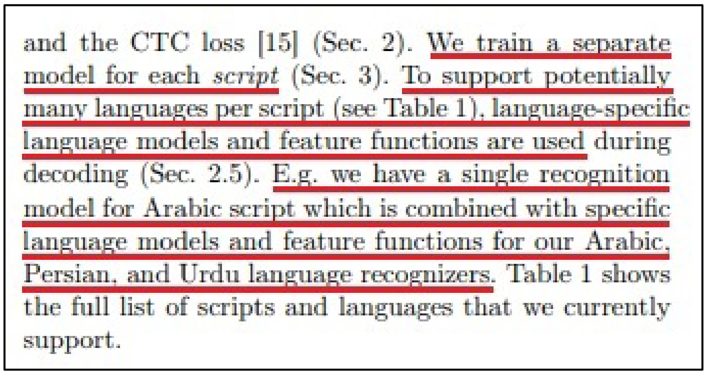
Personalization
Adapt Gboard to your typing and voice typing usage patterns

Improve voice and typing for everyone
Gboard makes model improvements on your device based on your usage patterns. With your permission, Gboard will use these improvements, in the aggregate, to update Google's voice and typing services

Learn more

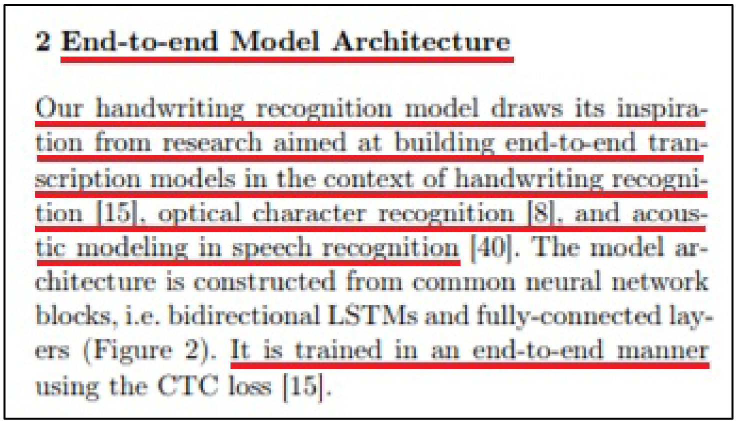
Delete learned words and data
Clear all on-device data that Gboard has saved to improve your typing and voice typing experience

Source: Screenshot taken during testing.



and the CTC loss [15] (Sec. 2). We train a separate model for each *script* (Sec. 3). To support potentially many languages per script (see Table 1), language-specific language models and feature functions are used during decoding (Sec. 2.5). E.g. we have a single recognition model for Arabic script which is combined with specific language models and feature functions for our Arabic, Persian, and Urdu language recognizers. Table 1 shows the full list of scripts and languages that we currently support.

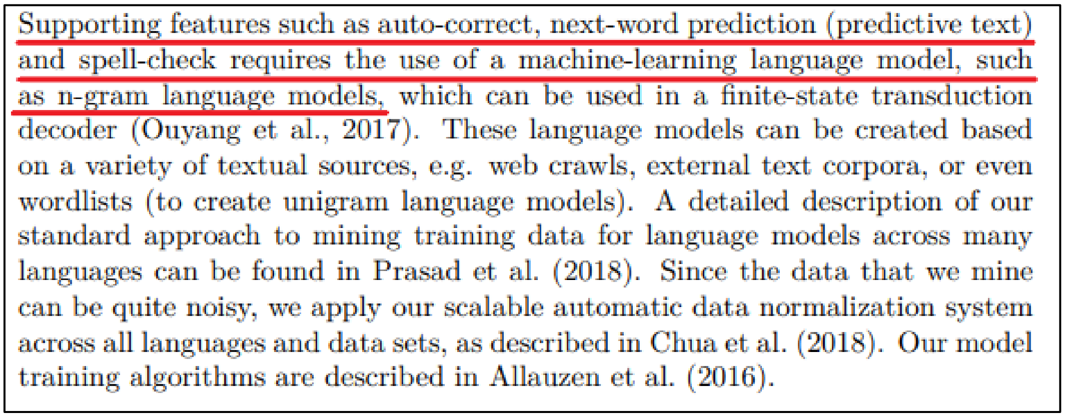
Source: https://arxiv.org/pdf/1902.10525.pdf (Page 2)



2 End-to-end Model Architecture

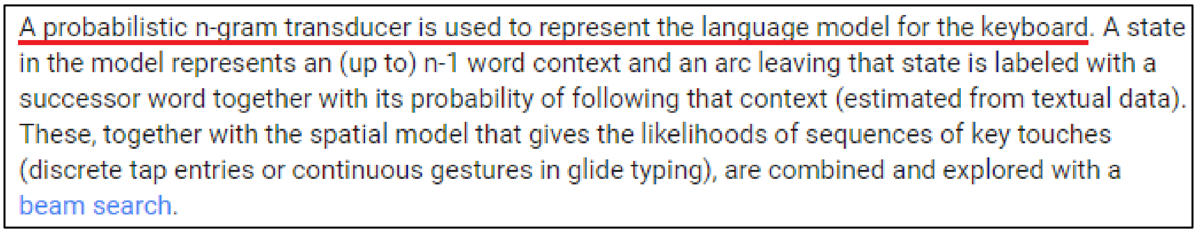
Our handwriting recognition model draws its inspiration from research aimed at building end-to-end transcription models in the context of handwriting recognition [15], optical character recognition [8], and acoustic modeling in speech recognition [40]. The model architecture is constructed from common neural network blocks, i.e. bidirectional LSTMs and fully-connected layers (Figure 2). It is trained in an end-to-end manner using the CTC loss [15].

Source: https://arxiv.org/pdf/1902.10525.pdf (Page 3)



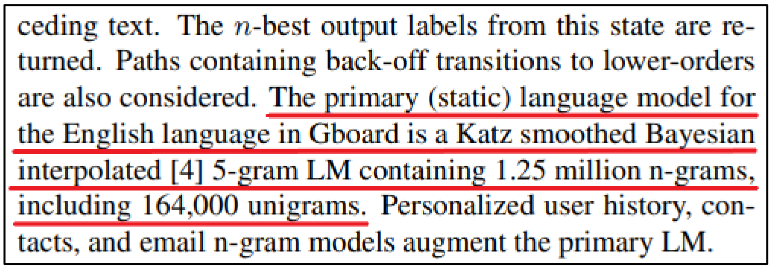
Supporting features such as auto-correct, next-word prediction (predictive text) and spell-check requires the use of a machine-learning language model, such as n-gram language models, which can be used in a finite-state transduction decoder (Ouyang et al., 2017). These language models can be created based on a variety of textual sources, e.g. web crawls, external text corpora, or even wordlists (to create unigram language models). A detailed description of our standard approach to mining training data for language models across many languages can be found in Prasad et al. (2018). Since the data that we mine can be quite noisy, we apply our scalable automatic data normalization system across all languages and data sets, as described in Chua et al. (2018). Our model training algorithms are described in Allauzen et al. (2016).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 16)



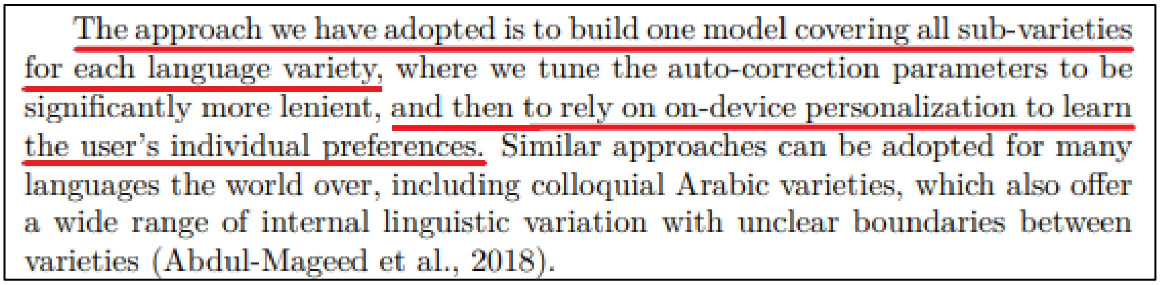
A probabilistic n-gram transducer is used to represent the language model for the keyboard. A state in the model represents an (up to) n-1 word context and an arc leaving that state is labeled with a successor word together with its probability of following that context (estimated from textual data). These, together with the spatial model that gives the likelihoods of sequences of key touches (discrete tap entries or continuous gestures in glide typing), are combined and explored with a beam search.

Source: https://ai.googleblog.com/2017/05/the-machine-intelligence-behind-gboard.html



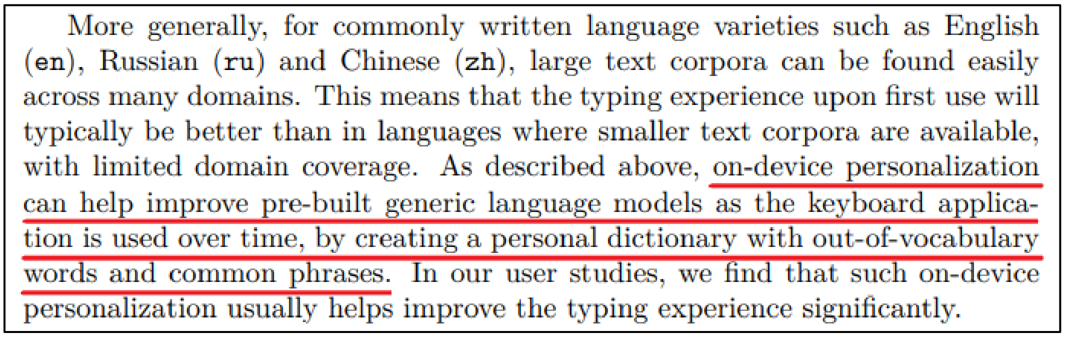
ceding text. The n-best output labels from this state are returned. Paths containing back-off transitions to lower-orders are also considered. The primary (static) language model for the English language in Gboard is a Katz smoothed Bayesian interpolated [4] 5-gram LM containing 1.25 million n-grams, including 164,000 unigrams. Personalized user history, contacts, and email n-gram models augment the primary LM.

Source: https://arxiv.org/pdf/1811.03604.pdf (Page 1)



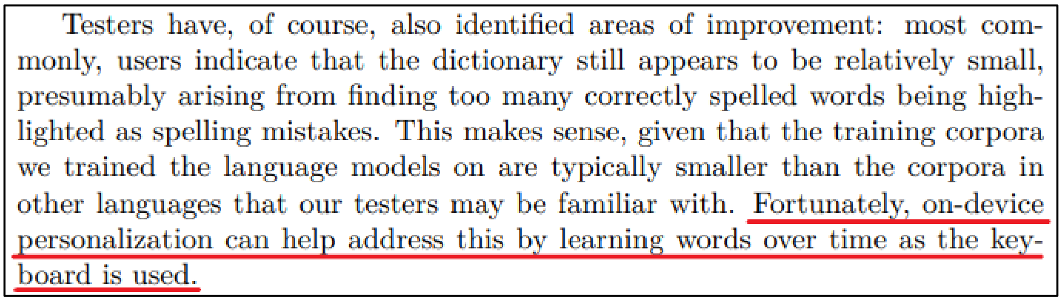
The approach we have adopted is to build one model covering all sub-varieties for each language variety, where we tune the auto-correction parameters to be significantly more lenient, and then to rely on on-device personalization to learn the user's individual preferences. Similar approaches can be adopted for many languages the world over, including colloquial Arabic varieties, which also offer a wide range of internal linguistic variation with unclear boundaries between varieties (Abdul-Mageed et al., 2018).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 17)



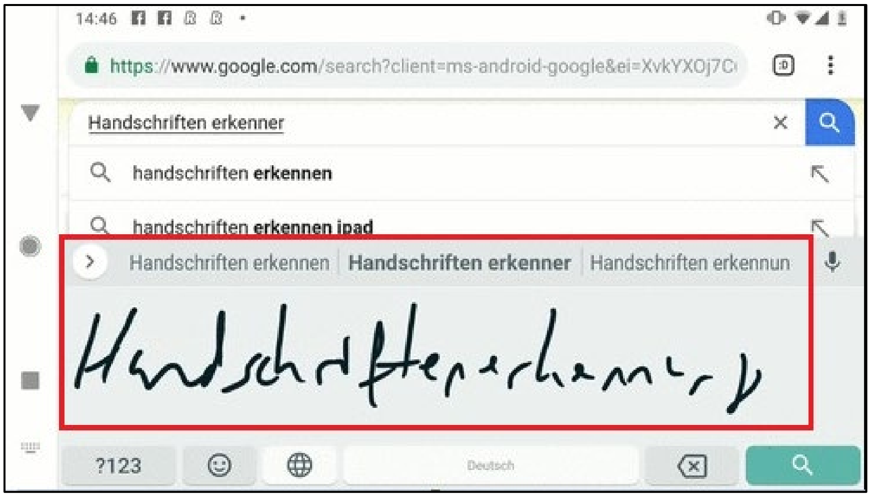
More generally, for commonly written language varieties such as English (en), Russian (ru) and Chinese (zh), large text corpora can be found easily across many domains. This means that the typing experience upon first use will typically be better than in languages where smaller text corpora are available, with limited domain coverage. As described above, on-device personalization can help improve pre-built generic language models as the keyboard application is used over time, by creating a personal dictionary with out-of-vocabulary words and common phrases. In our user studies, we find that such on-device personalization usually helps improve the typing experience significantly.

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 18)



Testers have, of course, also identified areas of improvement: most commonly, users indicate that the dictionary still appears to be relatively small, presumably arising from finding too many correctly spelled words being highlighted as spelling mistakes. This makes sense, given that the training corpora we trained the language models on are typically smaller than the corpora in other languages that our testers may be familiar with. Fortunately, on-device personalization can help address this by learning words over time as the keyboard is used.

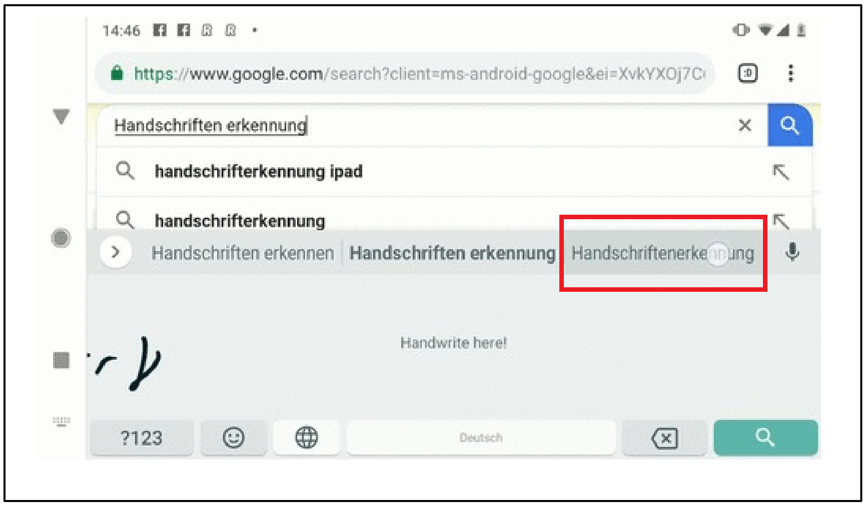
Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 19)



https://www.google.com/search?client=ms-android-google&ei=XvkYXOj7C

Handschriften erkenner

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html



https://www.google.com/search?client=ms-android-google&ei=XvkYXOj7C

Handschriften erkennung

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html

46. Buffalo Patents has been damaged as a result of the infringing conduct by ZTE alleged above. Thus, ZTE is liable to Buffalo Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

47. Buffalo Patents is only asserting method claims for the '405 Patent, so 35 U.S.C § 287(a) does not apply.

COUNT III

DIRECT INFRINGEMENT OF U.S. PATENT NO. 8,204,737

48. On June 19, 2012, the ’737 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Message Recognition Using Shared Language Model.”

49. Buffalo Patents is the owner of the ’737 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the ’737 Patent against infringers, and to collect damages for all relevant times.

50. ZTE made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, its ZTE Blade A7 Prime smartphone and other products[5] that include the Gboard application or similar virtual keyboard technology (“accused products”):

[5] *See, e.g.*, ZTE AVID (multiple models), ZTE Awe, ZTE AXON (multiple models), ZTE Blade (multiple models), ZTE Citrine LTE, ZTE Compel GoPhone, ZTE Cymbal-T, ZTE Fanfare, ZTE Fanfare 3, ZTE Flame, ZTE Fury, ZTE Quartz, ZTE Gabb (Z1, Z2), ZTE Grand (multiple models), ZTE Hawkeye, ZTE Imperial Max, ZTE JASPER, ZTE Libero 2, ZTE Majesty (Pro, Pro Plus), ZTE Maven (2, 3), ZTE Max (Blue LTE, Duo LTE, XL), ZTE NUBIA (multiple models), ZTE Obsidian, ZTE Overture 3, ZTE Prelude, ZTE Prelude 2, ZTE Prelude+, ZTE Prestige 2, ZTE Quest 5, ZTE S30, ZTE S30 Pro, ZTE S30 SE, ZTE small Fresh (4, 5, 5s), ZTE Sonata 3, ZTE Speed, ZTE Tempo, ZTE Tempo Go, ZTE Tempo X, ZTE Tough Max 2, ZTE V870, ZTE Visible R2, ZTE Vision R2, ZTE Voyage 4, ZTE Voyage 4S, ZTE Warp 7, ZTE Whirl 2, ZTE Z557, ZTE Z667 GoPhone, ZTE Zfive (multiple models), ZTE Zinger, ZTE ZMAX (multiple models), ZTE K88 Trek 2, ZTE K92 Primetime, ZTE ZPad (8”, 10”), ZTE V9, ZTE Gabb Smart Watch, ZTE Quartz, ZTE Spro, ZTE Spro 2, ZTE Spro Plus, etc.

Source: https://zteusa.com/products/zte-blade-a7-prime

51. By doing so, ZTE has directly infringed (literally and/or under the doctrine of equivalents) at least Claim 13 of the ’737 Patent. ZTE’s infringement in this regard is ongoing.

52. The ZTE Blade A7 Prime smartphone is an exemplary accused product.

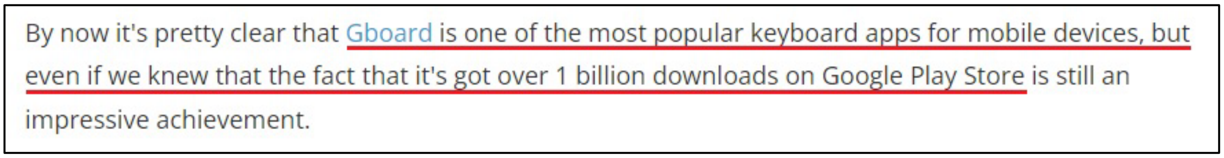
53. The ZTE Blade A7 Prime smartphone implements a method that includes the step of receiving, at a device, freehand input based at least in part on manipulation of a stylus.

54. For example, the ZTE Blade A7 Prime smartphone includes Gboard, which is the smartphone’s default virtual keyboard application. The virtual keyboard application on the ZTE Blade A7 Prime smartphone allows users to handwrite on the keyboard (“receiving, at a device, freehand input”) by providing an empty space upon which users can handwrite text using their finger or a stylus (“based at least in part on manipulation of a stylus”).

[add stylus evidence]

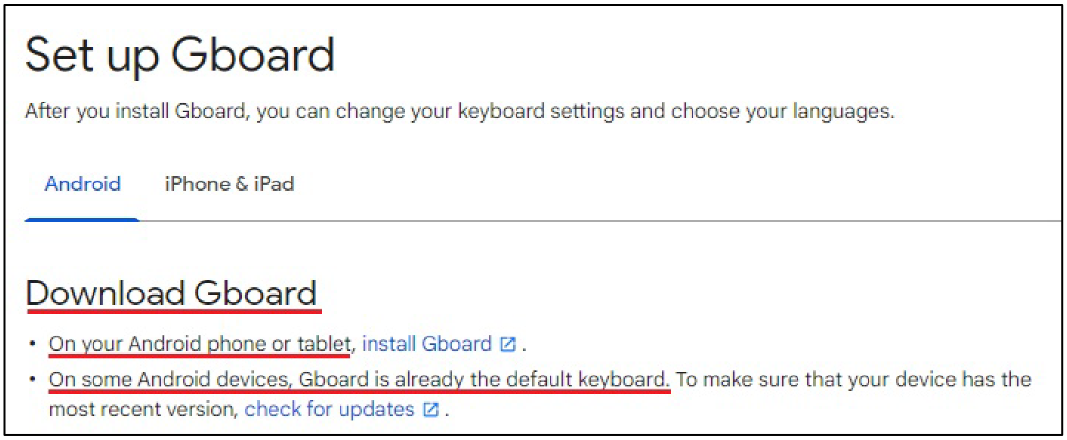
Stylus Pen for ZTE Blade A7 Prime (Stylus Pen by BoxWave) - FineTouch Capacitive Stylus, Super Precise Stylus Pen for ZTE Blade A7 Prime - Jet Black

Source: https://www.amazon.com/Stylus-BoxWave-FineTouch-Capacitive-Precise/dp/B081VVQ7NG



By now it's pretty clear that Gboard is one of the most popular keyboard apps for mobile devices, but even if we knew that the fact that it's got over 1 billion downloads on Google Play Store is still an impressive achievement.

Source: https://www.phonearena.com/news/Google-Gboard-keyboard-app-1-billion-downloads_id108061



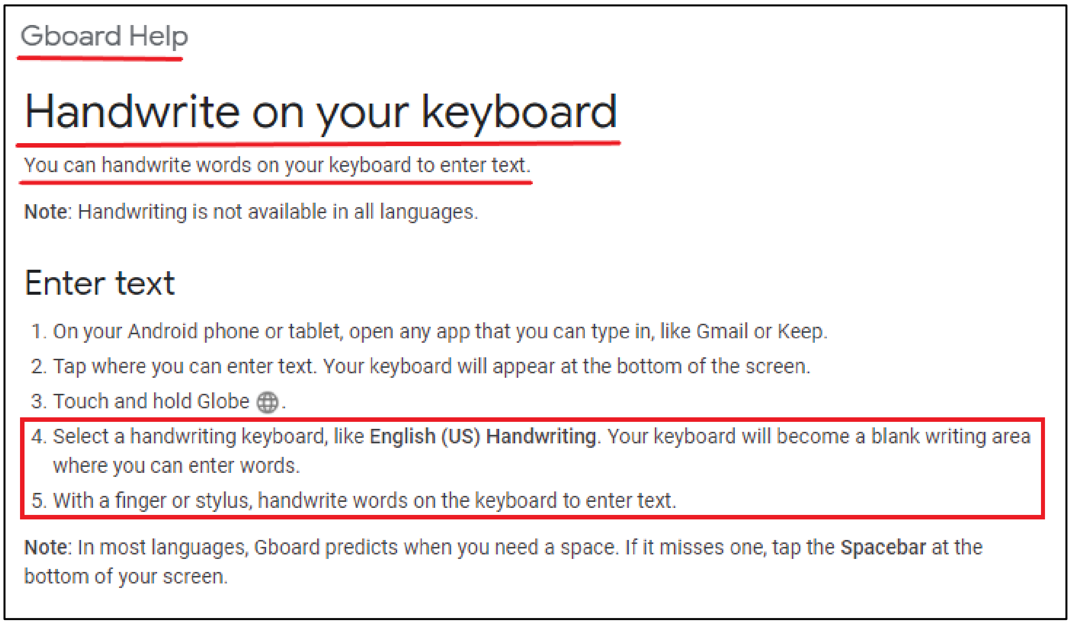
Set up Gboard

After you install Gboard, you can change your keyboard settings and choose your languages.

Download Gboard

- On your Android phone or tablet, install Gboard.
- On some Android devices, Gboard is already the default keyboard. To make sure that your device has the most recent version, check for updates.

Source:
https://support.google.com/gboard/answer/6380730?hl=en&co=GENIE.Platform%3DAndroid



Gboard Help

Handwrite on your keyboard

You can handwrite words on your keyboard to enter text.

Note: Handwriting is not available in all languages.

Enter text

1. On your Android phone or tablet, open any app that you can type in, like Gmail or Keep.
2. Tap where you can enter text. Your keyboard will appear at the bottom of the screen.
3. Touch and hold Globe 🌐.
4. Select a handwriting keyboard, like **English (US) Handwriting**. Your keyboard will become a blank writing area where you can enter words.
5. With a finger or stylus, handwrite words on the keyboard to enter text.

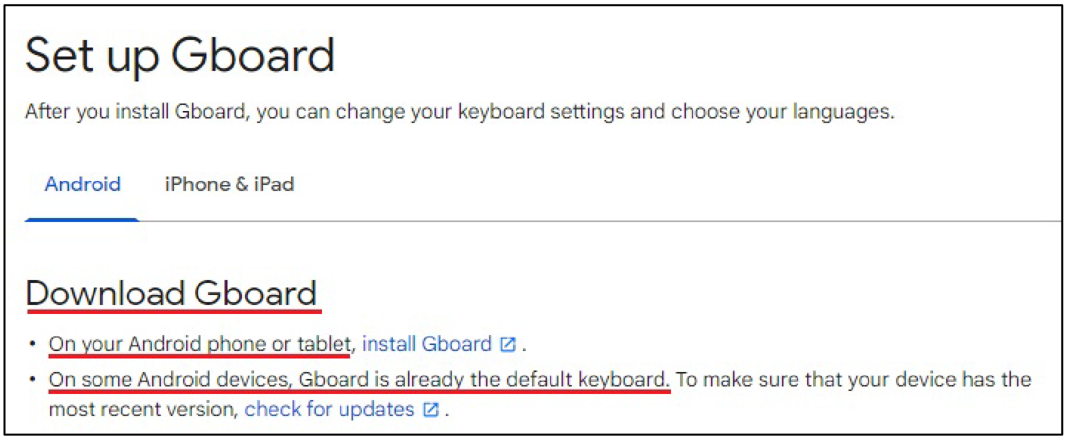
Note: In most languages, Gboard predicts when you need a space. If it misses one, tap the **Spacebar** at the bottom of your screen.

Source: https://support.google.com/gboard/answer/9108773?hl=en&ref_topic=9024098

55. The method implemented by the ZTE Blade A7 Prime smartphone includes the step of receiving, at the device, audio input based at least in part on information received at a microphone.

56. For example, the ZTE Blade A7 Prime smartphone includes Gboard, which is the smartphone's default virtual keyboard application. The virtual keyboard application on the ZTE Blade A7 Prime smartphone allows users to type with their voice ("information received at a

microphone"). When a user holds the microphone button and talks, the device receives the audio input ("receiving, at the device, audio input") and converts it to text.



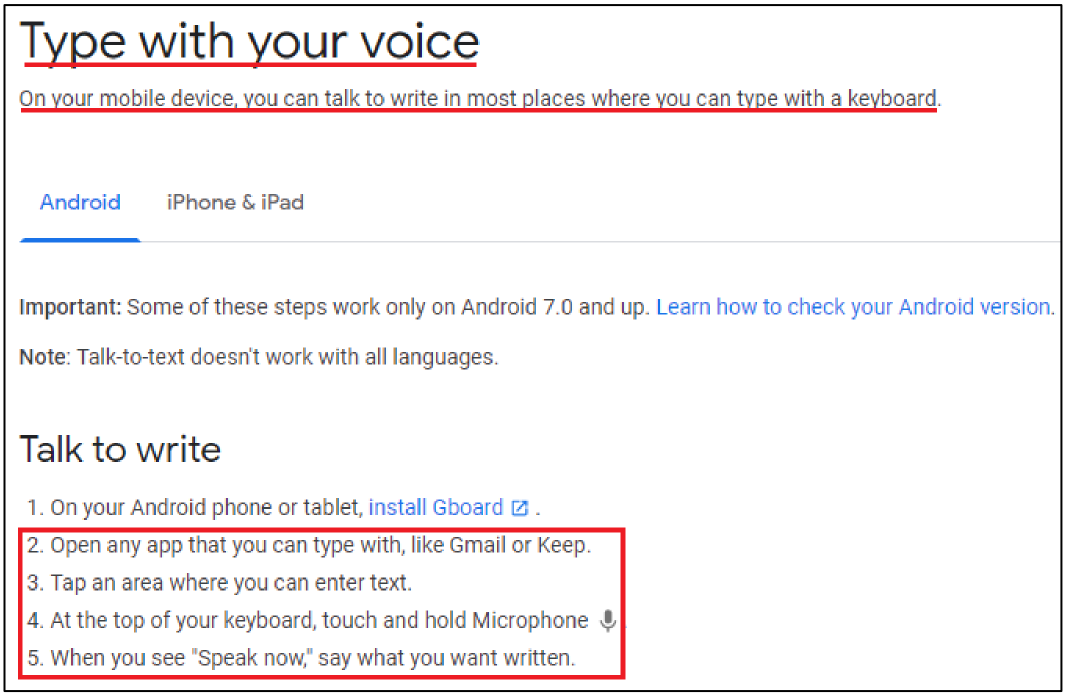
Set up Gboard

After you install Gboard, you can change your keyboard settings and choose your languages.

Download Gboard

- On your Android phone or tablet, install Gboard .
- On some Android devices, Gboard is already the default keyboard. To make sure that your device has the most recent version, check for updates .

Source:
https://support.google.com/gboard/answer/6380730?hl=en&co=GENIE.Platform%3DAndroid



Type with your voice

On your mobile device, you can talk to write in most places where you can type with a keyboard.

Important: Some of these steps work only on Android 7.0 and up. Learn how to check your Android version.

Note: Talk-to-text doesn't work with all languages.

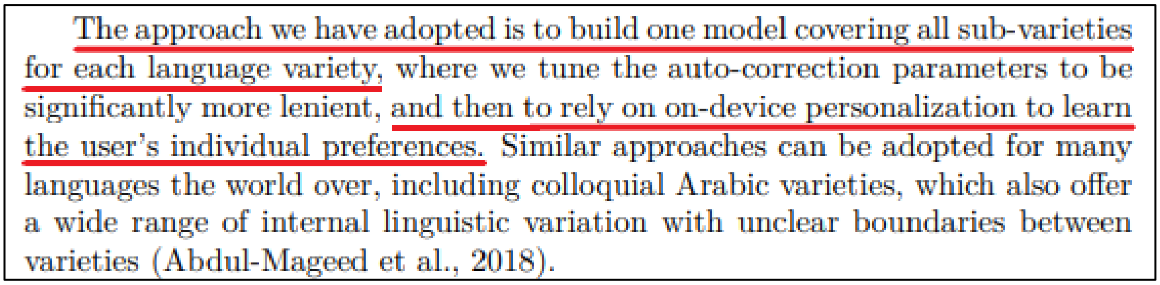
Talk to write

1. On your Android phone or tablet, install Gboard .
2. Open any app that you can type with, like Gmail or Keep.
3. Tap an area where you can enter text.
4. At the top of your keyboard, touch and hold Microphone 🎤.
5. When you see "Speak now," say what you want written.

Source: https://support.google.com/gboard/answer/2781851?hl=en&ref_topic=9024098

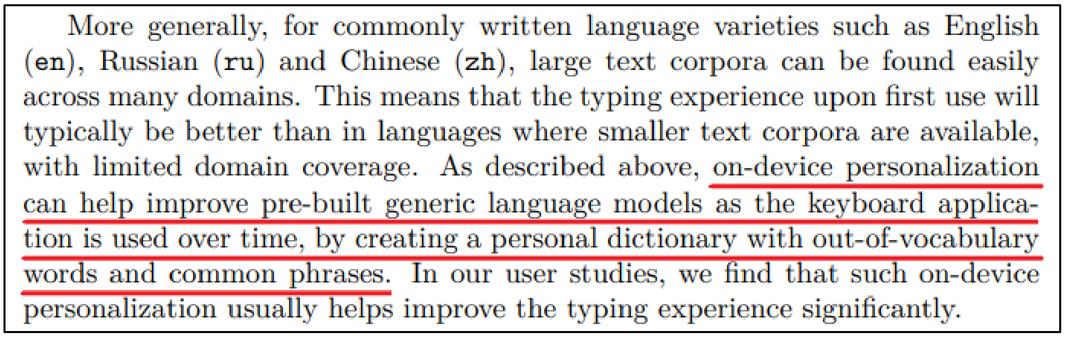
57. The method implemented by the ZTE Blade A7 Prime smartphone includes the step of selecting, via the device, a selected user message model from a plurality of user message models.

58. For example, the ZTE Blade A7 Prime smartphone includes Gboard, which is the smartphone's default virtual keyboard application. The virtual keyboard application on the ZTE Blade A7 Prime smartphone provides users with a personalized model to learn the user's preferences. As one example, users can select a profile ("selected user message model") from available profiles to be used as a virtual keyboard model. Examples of available profiles include personal and work profiles ("plurality of user message models").



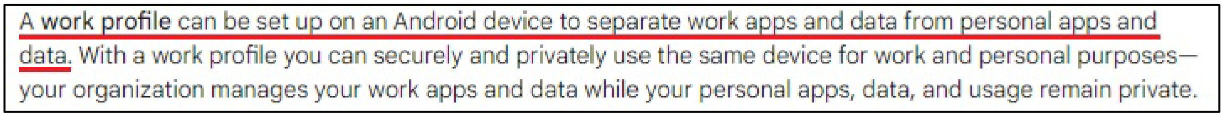
The approach we have adopted is to build one model covering all sub-varieties for each language variety, where we tune the auto-correction parameters to be significantly more lenient, and then to rely on on-device personalization to learn the user's individual preferences. Similar approaches can be adopted for many languages the world over, including colloquial Arabic varieties, which also offer a wide range of internal linguistic variation with unclear boundaries between varieties (Abdul-Mageed et al., 2018).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 17)



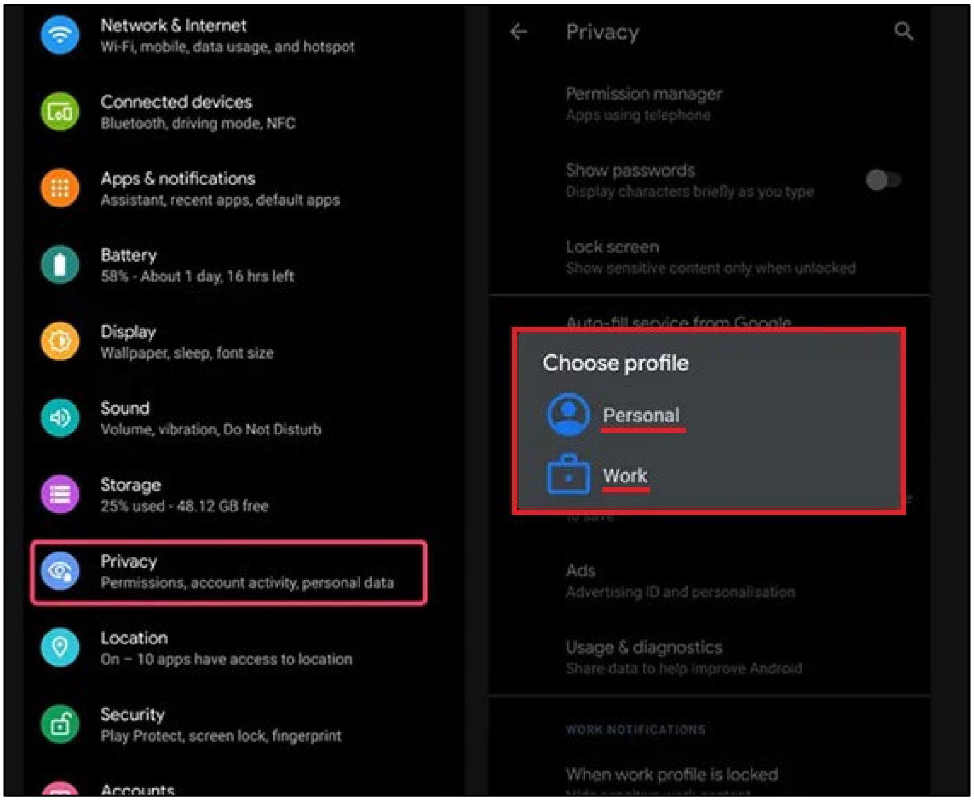
More generally, for commonly written language varieties such as English (en), Russian (ru) and Chinese (zh), large text corpora can be found easily across many domains. This means that the typing experience upon first use will typically be better than in languages where smaller text corpora are available, with limited domain coverage. As described above, on-device personalization can help improve pre-built generic language models as the keyboard application is used over time, by creating a personal dictionary with out-of-vocabulary words and common phrases. In our user studies, we find that such on-device personalization usually helps improve the typing experience significantly.

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 18)

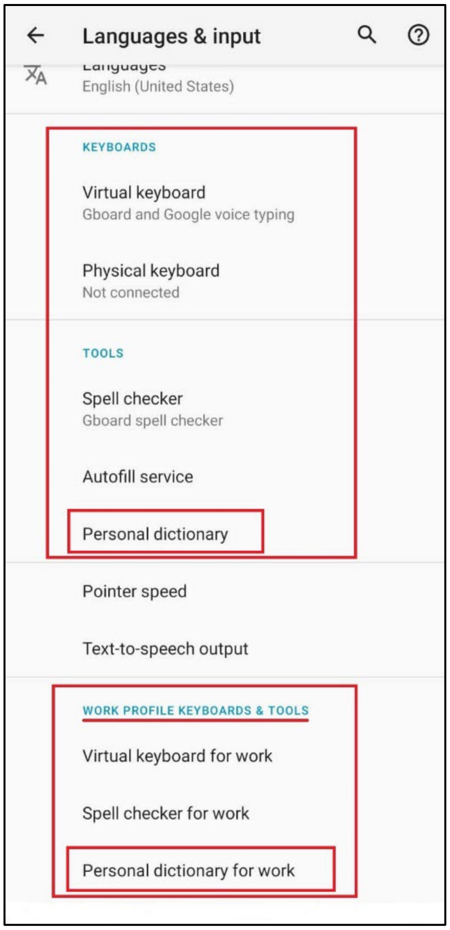


A work profile can be set up on an Android device to separate work apps and data from personal apps and data. With a work profile you can securely and privately use the same device for work and personal purposes—your organization manages your work apps and data while your personal apps, data, and usage remain private.

Source: https://support.google.com/work/android/answer/6191949


Network & Internet
Wi-Fi, mobile, data usage, and hotspot
Connected devices
Bluetooth, driving mode, NFC
Apps & notifications
Assistant, recent apps, default apps
Battery
58% - About 1 day, 16 hrs left
Display
Wallpaper, sleep, font size
Sound
Volume, vibration, Do Not Disturb
Storage
25% used - 48.12 GB free
Privacy
Permissions, account activity, personal data
Location
On – 10 apps have access to location
Security
Play Protect, screen lock, fingerprint
Accounts
Privacy
Permission manager
Apps using telephone
Show passwords
Display characters briefly as you type
Lock screen
Show sensitive content only when unlocked
Auto-fill service from Google
Choose profile
Personal
Work
Ads
Advertising ID and personalisation
Usage & diagnostics
Share data to help improve Android
WORK NOTIFICATIONS
When work profile is locked

Source: https://beebom.com/android-10-simplified-work-profiles/



Languages & input

Languages
English (United States)

KEYBOARDS

Virtual keyboard
Gboard and Google voice typing

Physical keyboard
Not connected

TOOLS

Spell checker
Gboard spell checker

Autofill service

Personal dictionary

Pointer speed

Text-to-speech output

WORK PROFILE KEYBOARDS & TOOLS

Virtual keyboard for work

Spell checker for work

Personal dictionary for work

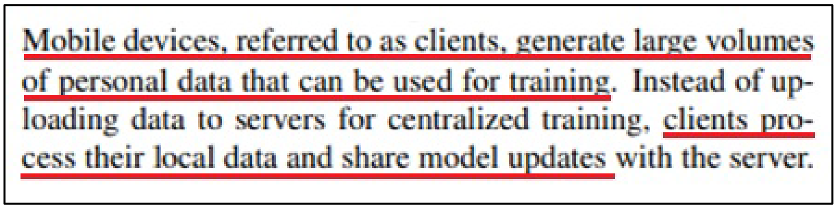
Source: Screenshot taken during testing.

59. The method implemented by the ZTE Blade A7 Prime smartphone includes the step of adjusting, via the device, a language model of a local message model, wherein the local message model includes the language model, a handwriting model, and an acoustic model, wherein the adjusting the language model is based at least in part on the selected user message model.

60. For example, the virtual keyboard application on the ZTE Blade A7 Prime smartphone provides users with a personalized model to learn the user's preferences. As one example, users can select a profile to be used as a virtual keyboard model from available profiles, such as personal and work profiles ("plurality of user message models"). After selecting a profile, the data input by the user (via the virtual keyboard) is used to learn the user's

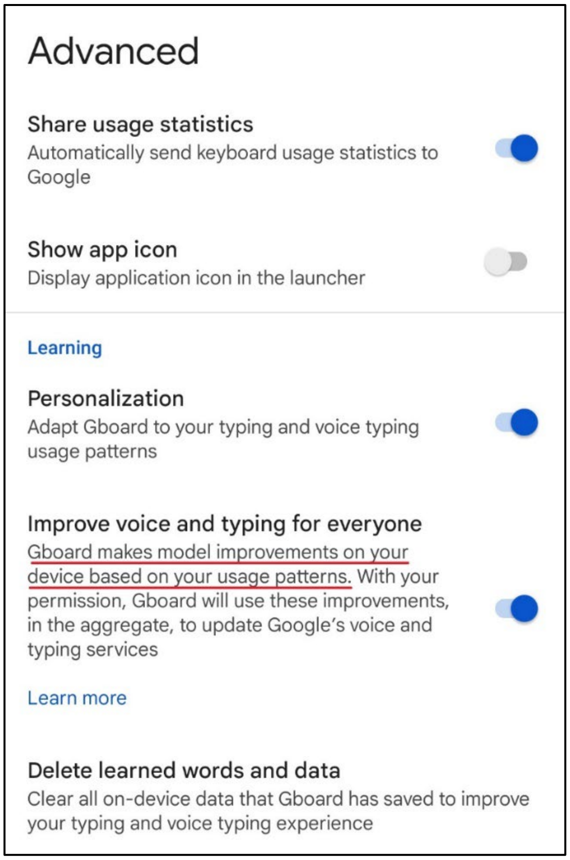
preferences, train the language model of the selected profile, and update the model locally. The smartphone builds one model for each language variety and uses on-device personalization to improve the language model of a selected profile by learning the user's preferences.

61. As one example, an end-to-end model architecture is used for the handwriting model and acoustic model. End-to-end models contain multiple functioning models tied together to form a single neural network that trains and improves itself. The end-to-end model ("local message model") contains an n-gram language model ("language model") personalized to the user, along with acoustic and handwriting models ("local message model includes the language model, a handwriting model, and an acoustic model"). Based on the input received from the user, the message correction data, and the selected profile of the user ("selected user message model"), the n-gram language model is trained and corrects itself to improve accuracy of prediction ("adjusting the language model").



Mobile devices, referred to as clients, generate large volumes of personal data that can be used for training. Instead of uploading data to servers for centralized training, clients process their local data and share model updates with the server.

Source: https://arxiv.org/pdf/1811.03604.pdf



Advanced

Share usage statistics
Automatically send keyboard usage statistics to Google

Show app icon
Display application icon in the launcher

Learning

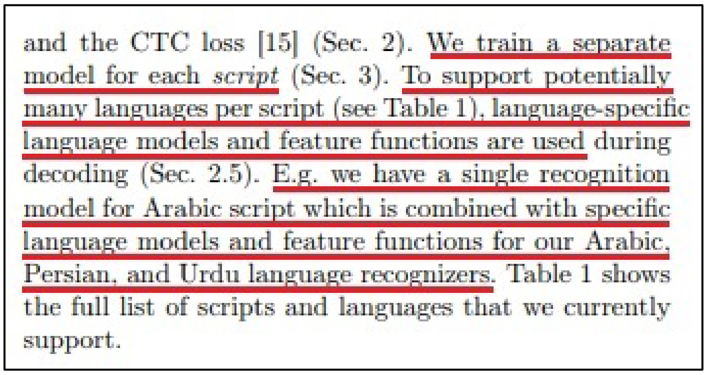
Personalization
Adapt Gboard to your typing and voice typing usage patterns

Improve voice and typing for everyone
Gboard makes model improvements on your device based on your usage patterns. With your permission, Gboard will use these improvements, in the aggregate, to update Google's voice and typing services

Learn more

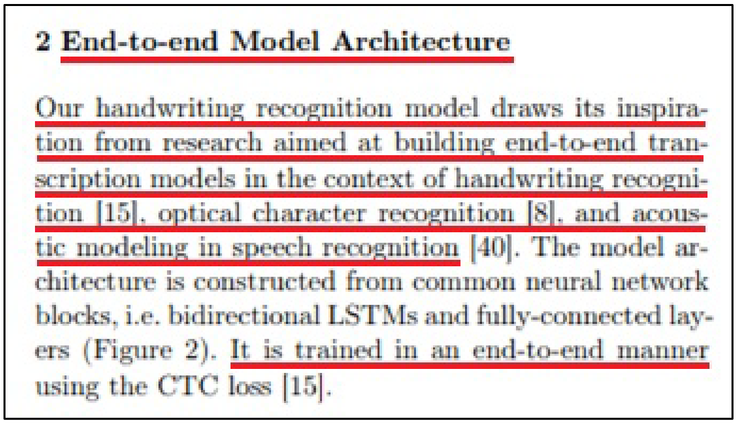
Delete learned words and data
Clear all on-device data that Gboard has saved to improve your typing and voice typing experience

Source: Screenshot taken during testing.



and the CTC loss [15] (Sec. 2). We train a separate model for each *script* (Sec. 3). To support potentially many languages per script (see Table 1), language-specific language models and feature functions are used during decoding (Sec. 2.5). E.g. we have a single recognition model for Arabic script which is combined with specific language models and feature functions for our Arabic, Persian, and Urdu language recognizers. Table 1 shows the full list of scripts and languages that we currently support.

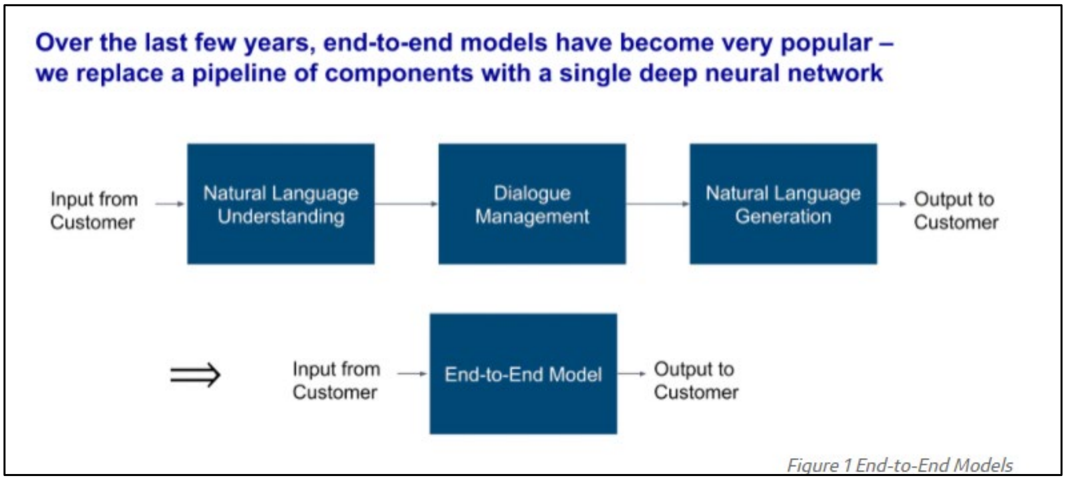
Source: https://arxiv.org/pdf/1902.10525.pdf (Page 2)



2 End-to-end Model Architecture

Our handwriting recognition model draws its inspiration from research aimed at building end-to-end transcription models in the context of handwriting recognition [15], optical character recognition [8], and acoustic modeling in speech recognition [40]. The model architecture is constructed from common neural network blocks, i.e. bidirectional LSTMs and fully-connected layers (Figure 2). It is trained in an end-to-end manner using the CTC loss [15].

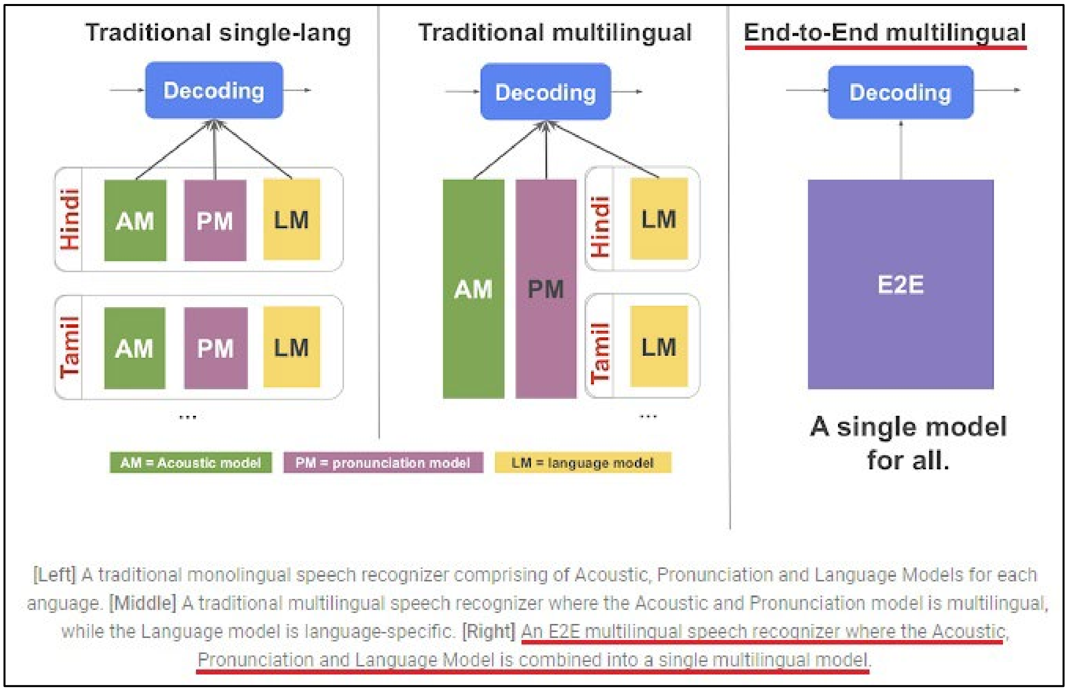
Source: https://arxiv.org/pdf/1902.10525.pdf (Page 3)



Over the last few years, end-to-end models have become very popular – we replace a pipeline of components with a single deep neural network

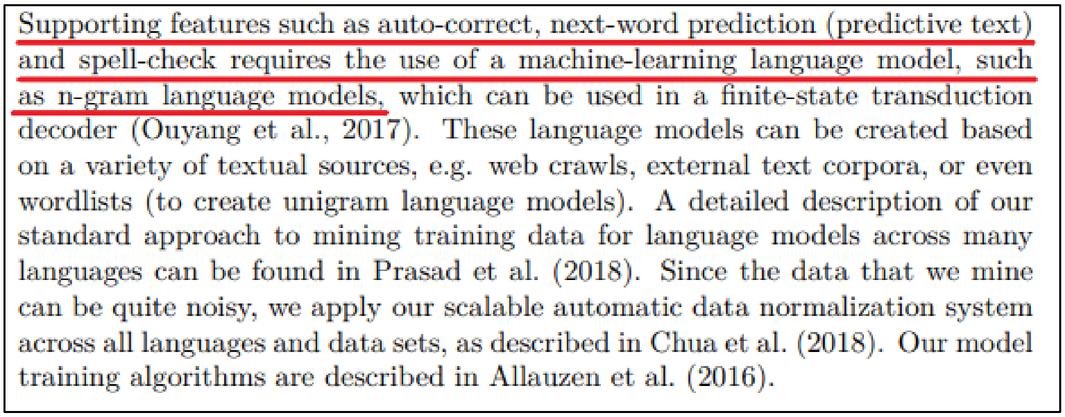
Figure 1 End-to-End Models

Source: https://www.capitalone.com/tech/machine-learning/pros-and-cons-of-end-to-end-models/



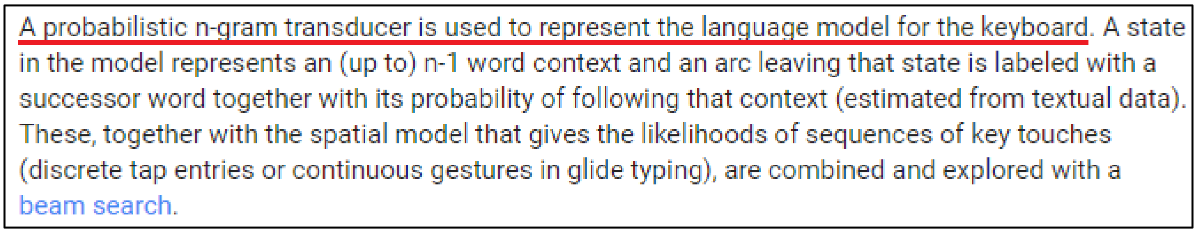
[Left] A traditional monolingual speech recognizer comprising of Acoustic, Pronunciation and Language Models for each anguage. [Middle] A traditional multilingual speech recognizer where the Acoustic and Pronunciation model is multilingual, while the Language model is language-specific. [Right] An E2E multilingual speech recognizer where the Acoustic, Pronunciation and Language Model is combined into a single multilingual model.

Source: https://ai.googleblog.com/2019/09/large-scale-multilingual-speech.html



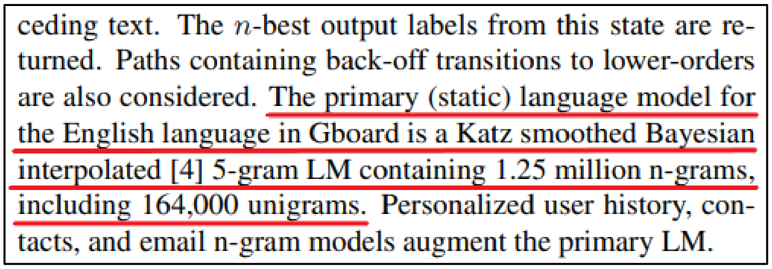
Supporting features such as auto-correct, next-word prediction (predictive text) and spell-check requires the use of a machine-learning language model, such as n-gram language models, which can be used in a finite-state transduction decoder (Ouyang et al., 2017). These language models can be created based on a variety of textual sources, e.g. web crawls, external text corpora, or even wordlists (to create unigram language models). A detailed description of our standard approach to mining training data for language models across many languages can be found in Prasad et al. (2018). Since the data that we mine can be quite noisy, we apply our scalable automatic data normalization system across all languages and data sets, as described in Chua et al. (2018). Our model training algorithms are described in Allauzen et al. (2016).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 16)



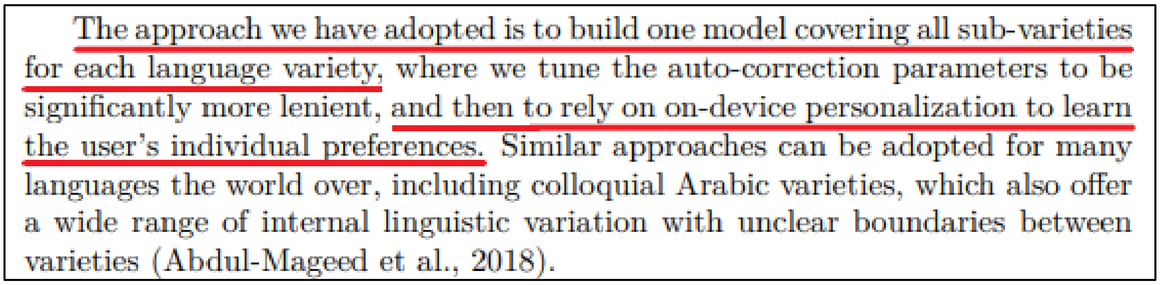
A probabilistic n-gram transducer is used to represent the language model for the keyboard. A state in the model represents an (up to) n-1 word context and an arc leaving that state is labeled with a successor word together with its probability of following that context (estimated from textual data). These, together with the spatial model that gives the likelihoods of sequences of key touches (discrete tap entries or continuous gestures in glide typing), are combined and explored with a beam search.

Source: https://ai.googleblog.com/2017/05/the-machine-intelligence-behind-gboard.html



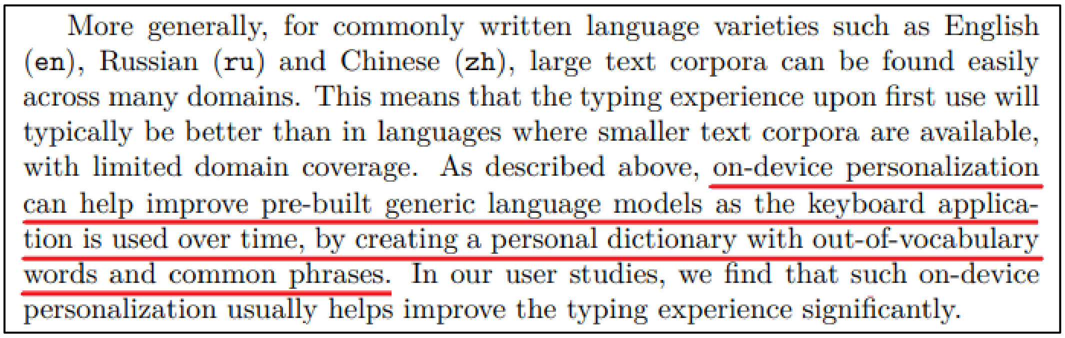
ceding text. The n-best output labels from this state are returned. Paths containing back-off transitions to lower-orders are also considered. The primary (static) language model for the English language in Gboard is a Katz smoothed Bayesian interpolated [4] 5-gram LM containing 1.25 million n-grams, including 164,000 unigrams. Personalized user history, contacts, and email n-gram models augment the primary LM.

Source: https://arxiv.org/pdf/1811.03604.pdf (Page 1)



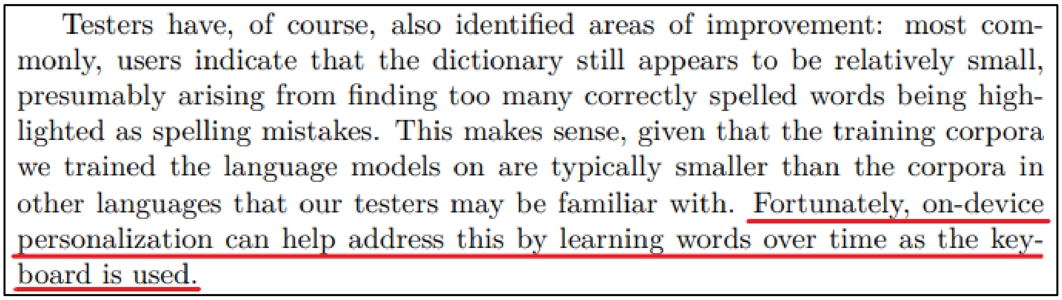
The approach we have adopted is to build one model covering all sub-varieties for each language variety, where we tune the auto-correction parameters to be significantly more lenient, and then to rely on on-device personalization to learn the user's individual preferences. Similar approaches can be adopted for many languages the world over, including colloquial Arabic varieties, which also offer a wide range of internal linguistic variation with unclear boundaries between varieties (Abdul-Mageed et al., 2018).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 17)



More generally, for commonly written language varieties such as English (`en`), Russian (`ru`) and Chinese (`zh`), large text corpora can be found easily across many domains. This means that the typing experience upon first use will typically be better than in languages where smaller text corpora are available, with limited domain coverage. As described above, on-device personalization can help improve pre-built generic language models as the keyboard application is used over time, by creating a personal dictionary with out-of-vocabulary words and common phrases. In our user studies, we find that such on-device personalization usually helps improve the typing experience significantly.

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 18)

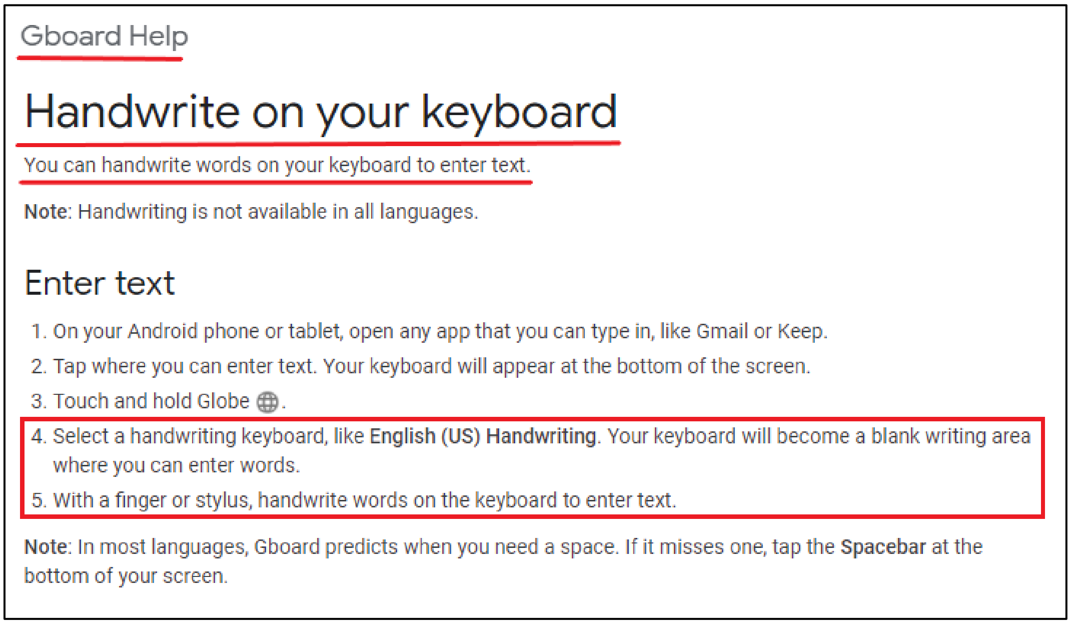


Testers have, of course, also identified areas of improvement: most commonly, users indicate that the dictionary still appears to be relatively small, presumably arising from finding too many correctly spelled words being highlighted as spelling mistakes. This makes sense, given that the training corpora we trained the language models on are typically smaller than the corpora in other languages that our testers may be familiar with. Fortunately, on-device personalization can help address this by learning words over time as the keyboard is used.

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 19)

62. The method implemented by the ZTE Blade A7 Prime smartphone includes the step of generating, by the device, converted handwriting text data based at least in part on the freehand input, the adjusted language model, and the handwriting model.

63. For example, the virtual keyboard application on the ZTE Blade A7 Prime smartphone provides a feature for converting handwriting to text. It uses an end-to-end model based on the Recurrent Neural Network (RNN), which combines handwriting and language models for handwriting recognition. The language model used by the smartphone is an n-gram language model. In the writing area of the keypad of the device, patterns created by the user are recorded, and the corresponding handwriting information ("freehand input") is converted into text using a handwriting model ("handwriting model"), which is a bi-directional version of the quasi-recurrent neural network (QRNN) model. The user-personalized language model ("adjusted language model") is used to generate text responsive to handwriting input ("converted handwriting text data").



Gboard Help

Handwrite on your keyboard

You can handwrite words on your keyboard to enter text.

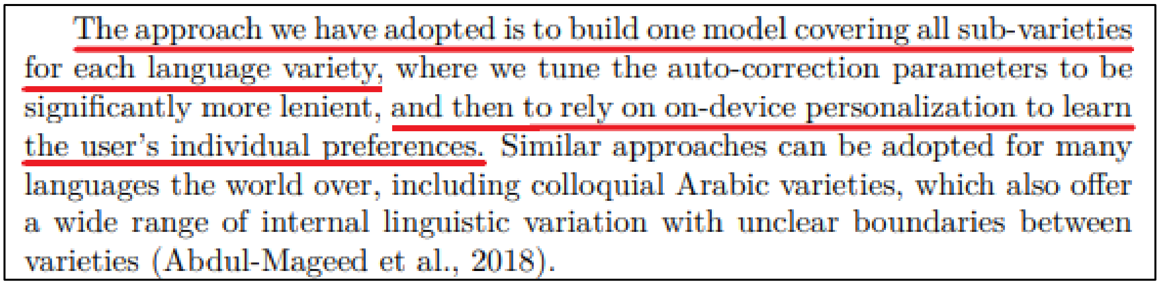
Note: Handwriting is not available in all languages.

Enter text

1. On your Android phone or tablet, open any app that you can type in, like Gmail or Keep.
2. Tap where you can enter text. Your keyboard will appear at the bottom of the screen.
3. Touch and hold Globe 🌐.
4. Select a handwriting keyboard, like **English (US) Handwriting**. Your keyboard will become a blank writing area where you can enter words.
5. With a finger or stylus, handwrite words on the keyboard to enter text.

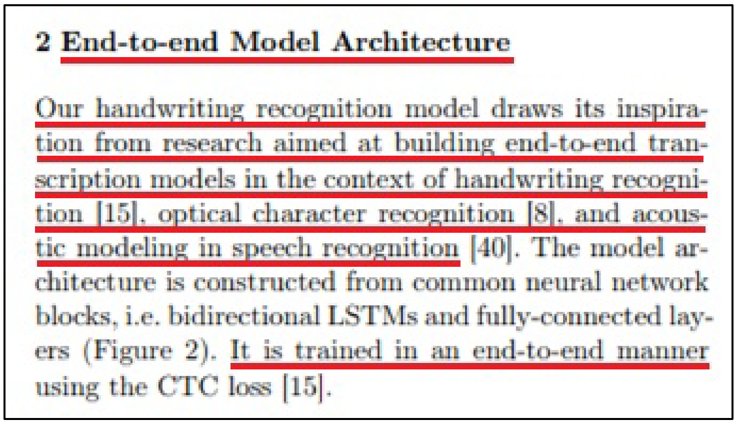
Note: In most languages, Gboard predicts when you need a space. If it misses one, tap the **Spacebar** at the bottom of your screen.

Source: https://support.google.com/gboard/answer/9108773?hl=en&ref_topic=9024098



The approach we have adopted is to build one model covering all sub-varieties for each language variety, where we tune the auto-correction parameters to be significantly more lenient, and then to rely on on-device personalization to learn the user's individual preferences. Similar approaches can be adopted for many languages the world over, including colloquial Arabic varieties, which also offer a wide range of internal linguistic variation with unclear boundaries between varieties (Abdul-Mageed et al., 2018).

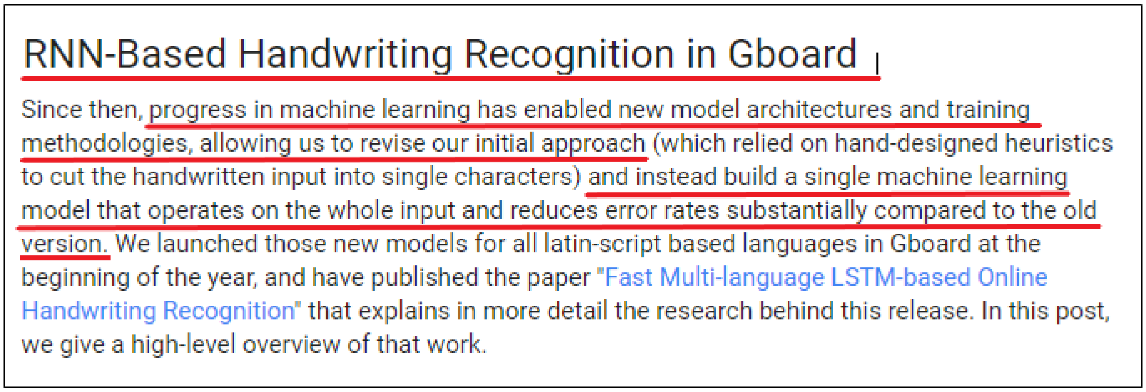
Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 17)



2 End-to-end Model Architecture

Our handwriting recognition model draws its inspiration from research aimed at building end-to-end transcription models in the context of handwriting recognition [15], optical character recognition [8], and acoustic modeling in speech recognition [40]. The model architecture is constructed from common neural network blocks, i.e. bidirectional LSTMs and fully-connected layers (Figure 2). It is trained in an end-to-end manner using the CTC loss [15].

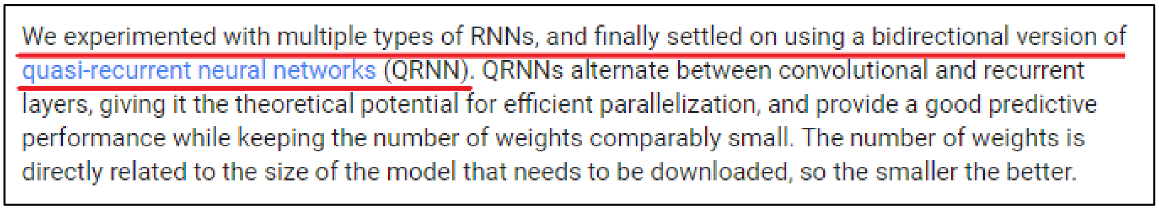
Source: https://arxiv.org/pdf/1902.10525.pdf (Page 3)



RNN-Based Handwriting Recognition in Gboard

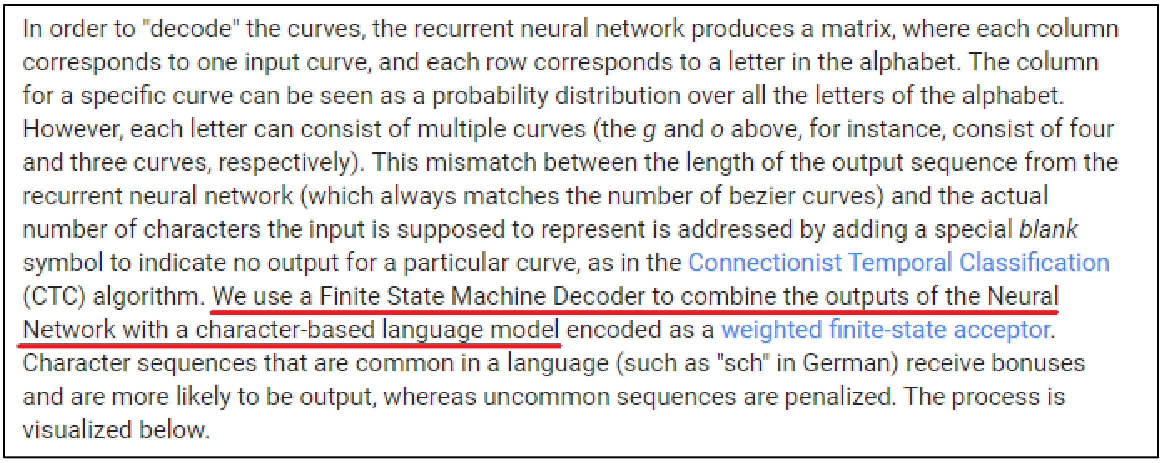
Since then, progress in machine learning has enabled new model architectures and training methodologies, allowing us to revise our initial approach (which relied on hand-designed heuristics to cut the handwritten input into single characters) and instead build a single machine learning model that operates on the whole input and reduces error rates substantially compared to the old version. We launched those new models for all latin-script based languages in Gboard at the beginning of the year, and have published the paper "Fast Multi-language LSTM-based Online Handwriting Recognition" that explains in more detail the research behind this release. In this post, we give a high-level overview of that work.

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html



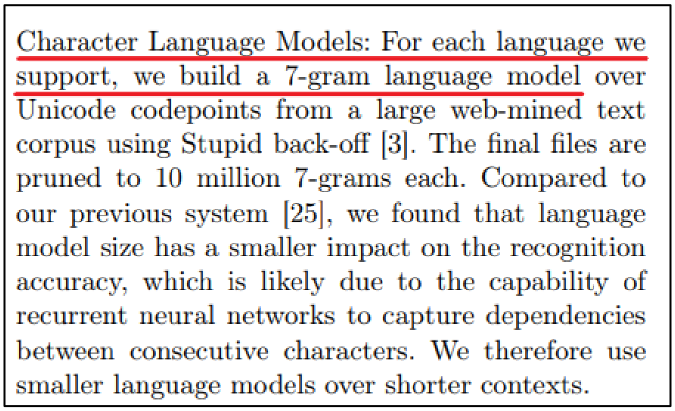
We experimented with multiple types of RNNs, and finally settled on using a bidirectional version of quasi-recurrent neural networks (QRNN). QRNNs alternate between convolutional and recurrent layers, giving it the theoretical potential for efficient parallelization, and provide a good predictive performance while keeping the number of weights comparably small. The number of weights is directly related to the size of the model that needs to be downloaded, so the smaller the better.

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html



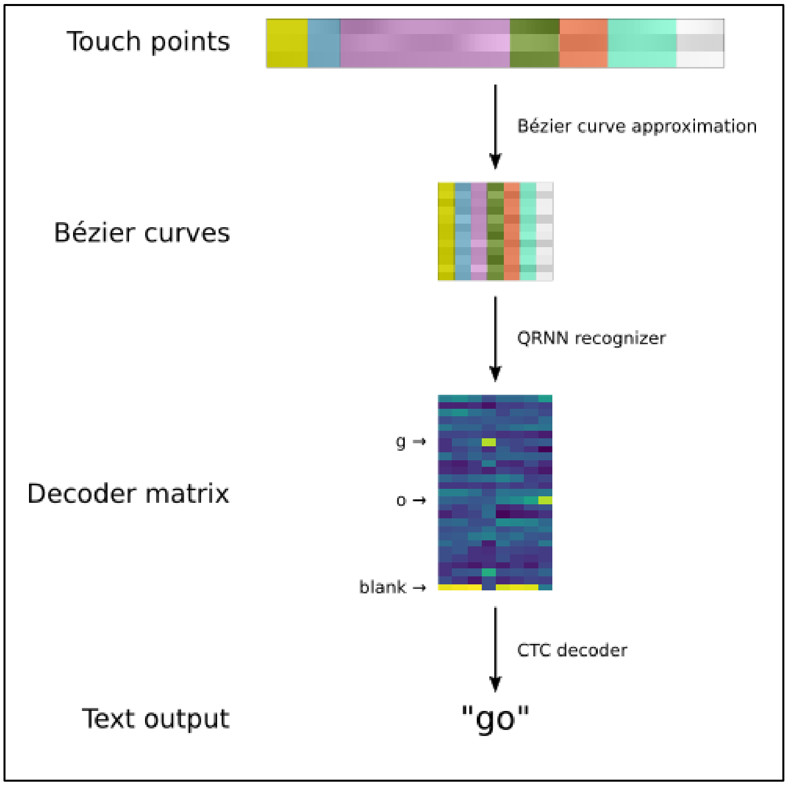
In order to "decode" the curves, the recurrent neural network produces a matrix, where each column corresponds to one input curve, and each row corresponds to a letter in the alphabet. The column for a specific curve can be seen as a probability distribution over all the letters of the alphabet. However, each letter can consist of multiple curves (the *g* and *o* above, for instance, consist of four and three curves, respectively). This mismatch between the length of the output sequence from the recurrent neural network (which always matches the number of bezier curves) and the actual number of characters the input is supposed to represent is addressed by adding a special *blank* symbol to indicate no output for a particular curve, as in the Connectionist Temporal Classification (CTC) algorithm. We use a Finite State Machine Decoder to combine the outputs of the Neural Network with a character-based language model encoded as a weighted finite-state acceptor. Character sequences that are common in a language (such as "sch" in German) receive bonuses and are more likely to be output, whereas uncommon sequences are penalized. The process is visualized below.

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html

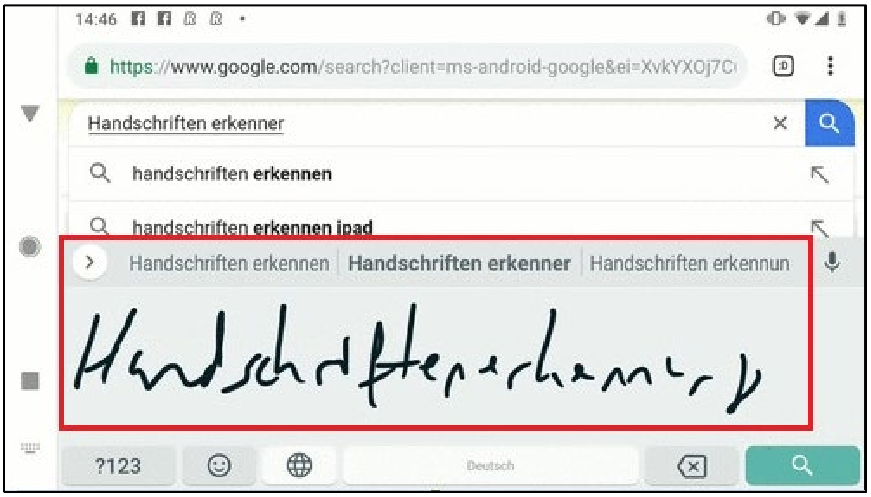


Character Language Models: For each language we support, we build a 7-gram language model over Unicode codepoints from a large web-mined text corpus using Stupid back-off [3]. The final files are pruned to 10 million 7-grams each. Compared to our previous system [25], we found that language model size has a smaller impact on the recognition accuracy, which is likely due to the capability of recurrent neural networks to capture dependencies between consecutive characters. We therefore use smaller language models over shorter contexts.

Source: https://arxiv.org/pdf/1902.10525.pdf



Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html


14:46

https://www.google.com/search?client=ms-android-google&ei=XvkYXOj7C

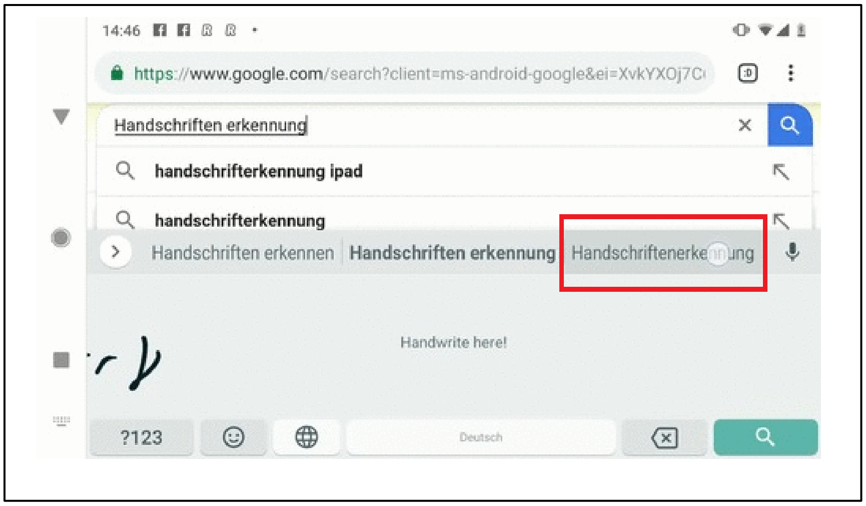
Handschriften erkenner

handschriften **erkennen**

handschriften **erkennen ipad**

Handschriften erkennen | **Handschriften erkenner** | Handschriften erkennun

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html



https://www.google.com/search?client=ms-android-google&ei=XvkYXOj7C

Handschriften erkennung

handschrifterkennung ipad

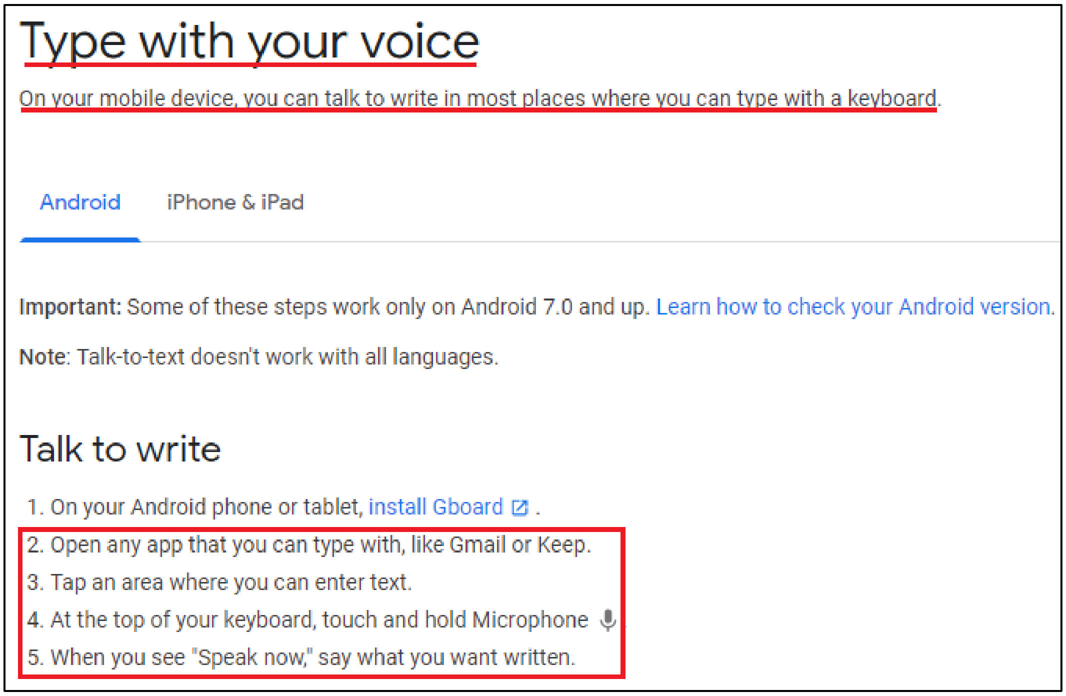
handschrifterkennung

Handschriften erkennen | Handschriften erkennung | Handschriftenerkennung

Source: https://ai.googleblog.com/2019/03/rnn-based-handwriting-recognition-in.html

64. The method implemented by the ZTE Blade A7 Prime smartphone includes the step of generating, by the device, converted speech text data based at least in part on the audio input, the adjusted language model, and the acoustic model.

65. For example, the virtual keyboard application on the ZTE Blade A7 Prime smartphone uses an on-device, all neural speech recognizer. The speech recognizer uses an end-to-end RNN-T model that combines an acoustic model, pronunciation model, and a language model. The language model is an n-gram language model. The user's voice input ("audio input") is converted into text using the speech recognizer. The pronunciation model, acoustic model ("acoustic model"), and the user-personalized language model ("adjusted language model") can generate text responsive to voice input ("generating, by the device, converted speech text data").



Type with your voice

On your mobile device, you can talk to write in most places where you can type with a keyboard.

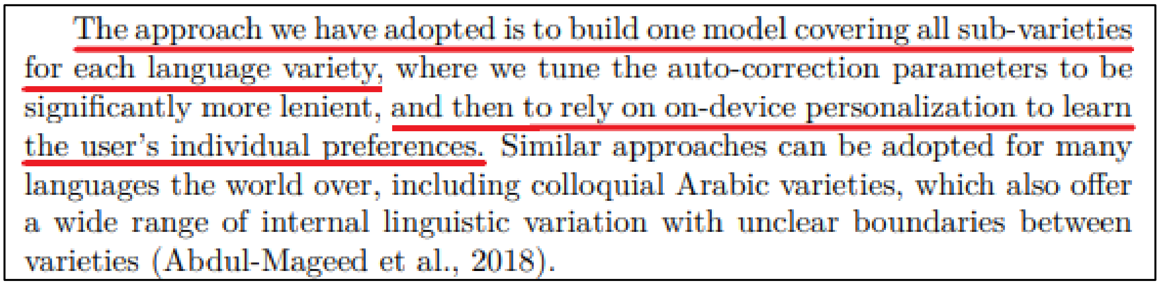
Important: Some of these steps work only on Android 7.0 and up. Learn how to check your Android version.

Note: Talk-to-text doesn't work with all languages.

Talk to write

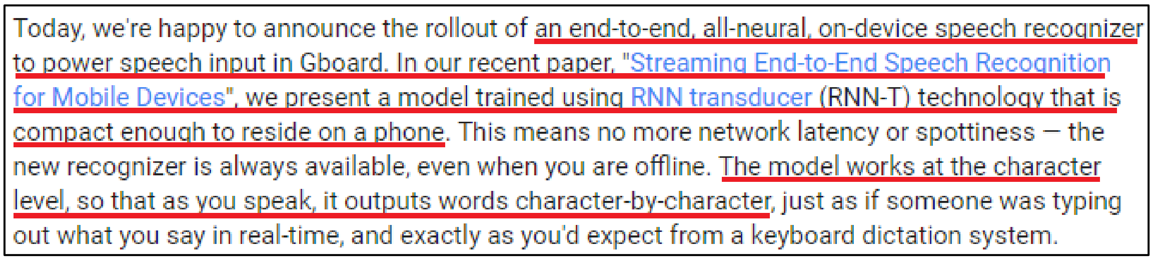
1. On your Android phone or tablet, install Gboard ↗ .
2. Open any app that you can type with, like Gmail or Keep.
3. Tap an area where you can enter text.
4. At the top of your keyboard, touch and hold Microphone 🎤.
5. When you see "Speak now," say what you want written.

Source: https://support.google.com/gboard/answer/2781851?hl=en&ref_topic=9024098



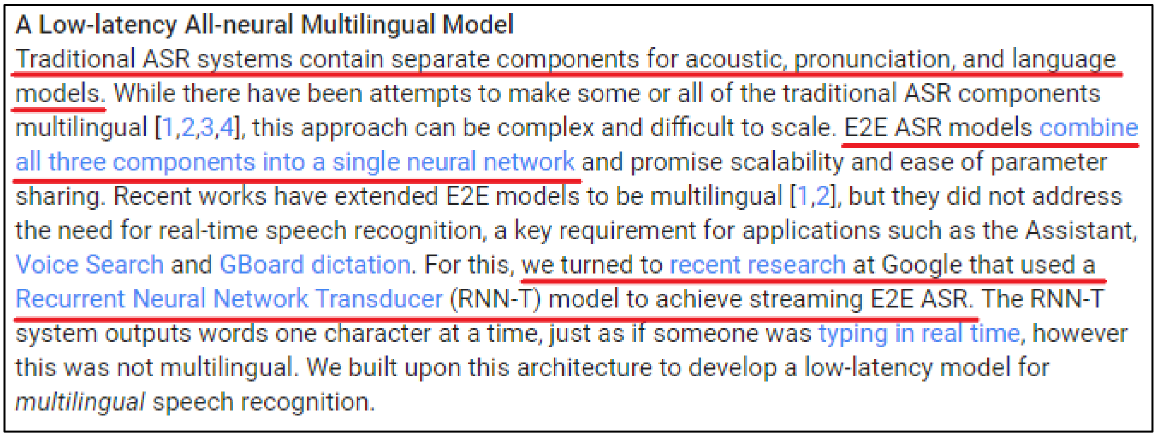
The approach we have adopted is to build one model covering all sub-varieties for each language variety, where we tune the auto-correction parameters to be significantly more lenient, and then to rely on on-device personalization to learn the user's individual preferences. Similar approaches can be adopted for many languages the world over, including colloquial Arabic varieties, which also offer a wide range of internal linguistic variation with unclear boundaries between varieties (Abdul-Mageed et al., 2018).

Source: https://arxiv.org/ftp/arxiv/papers/1912/1912.01218.pdf (Page 17)



Today, we're happy to announce the rollout of an end-to-end, all-neural, on-device speech recognizer to power speech input in Gboard. In our recent paper, "Streaming End-to-End Speech Recognition for Mobile Devices", we present a model trained using RNN transducer (RNN-T) technology that is compact enough to reside on a phone. This means no more network latency or spottiness — the new recognizer is always available, even when you are offline. The model works at the character level, so that as you speak, it outputs words character-by-character, just as if someone was typing out what you say in real-time, and exactly as you'd expect from a keyboard dictation system.

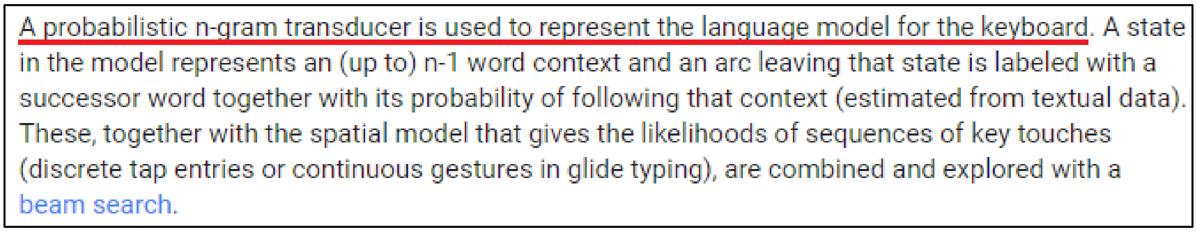
Source: https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html



A Low-latency All-neural Multilingual Model

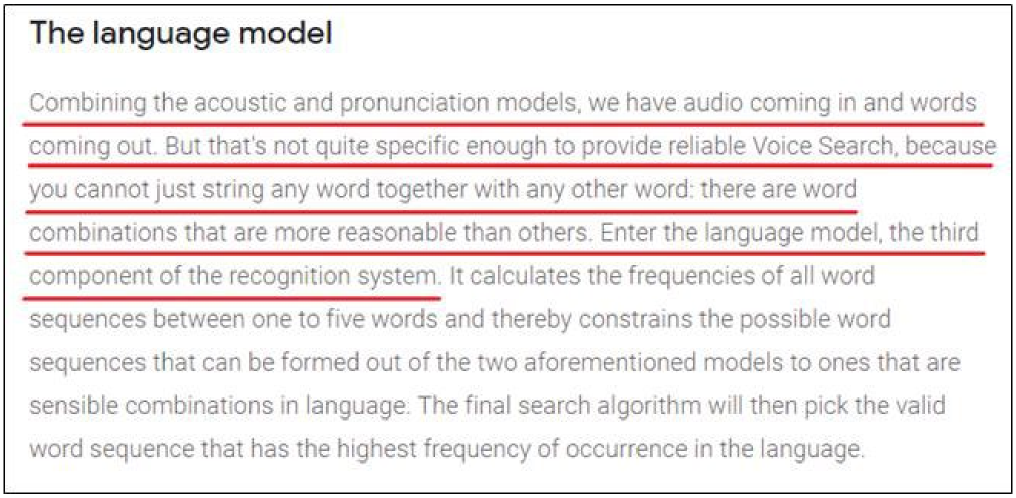
Traditional ASR systems contain separate components for acoustic, pronunciation, and language models. While there have been attempts to make some or all of the traditional ASR components multilingual [1,2,3,4], this approach can be complex and difficult to scale. E2E ASR models combine all three components into a single neural network and promise scalability and ease of parameter sharing. Recent works have extended E2E models to be multilingual [1,2], but they did not address the need for real-time speech recognition, a key requirement for applications such as the Assistant, Voice Search and GBoard dictation. For this, we turned to recent research at Google that used a Recurrent Neural Network Transducer (RNN-T) model to achieve streaming E2E ASR. The RNN-T system outputs words one character at a time, just as if someone was typing in real time, however this was not multilingual. We built upon this architecture to develop a low-latency model for *multilingual* speech recognition.

Source: https://ai.googleblog.com/2019/09/large-scale-multilingual-speech.html



A probabilistic n-gram transducer is used to represent the language model for the keyboard. A state in the model represents an (up to) n-1 word context and an arc leaving that state is labeled with a successor word together with its probability of following that context (estimated from textual data). These, together with the spatial model that gives the likelihoods of sequences of key touches (discrete tap entries or continuous gestures in glide typing), are combined and explored with a beam search.

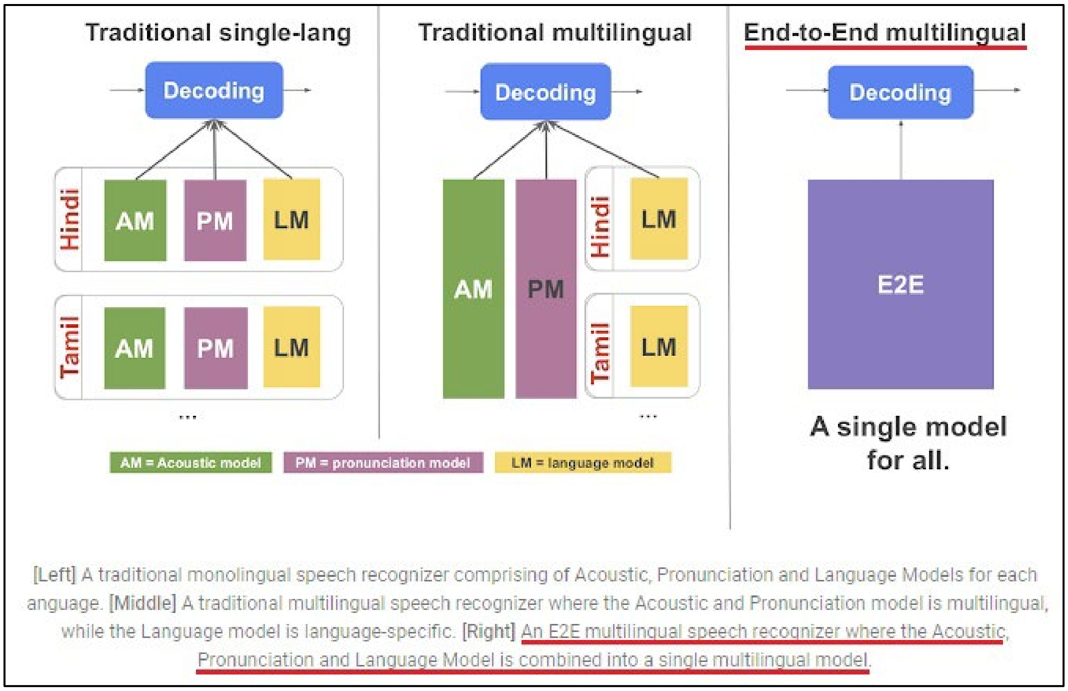
Source: https://ai.googleblog.com/2017/05/the-machine-intelligence-behind-gboard.html



The language model

Combining the acoustic and pronunciation models, we have audio coming in and words coming out. But that's not quite specific enough to provide reliable Voice Search, because you cannot just string any word together with any other word: there are word combinations that are more reasonable than others. Enter the language model, the third component of the recognition system. It calculates the frequencies of all word sequences between one to five words and thereby constrains the possible word sequences that can be formed out of the two aforementioned models to ones that are sensible combinations in language. The final search algorithm will then pick the valid word sequence that has the highest frequency of occurrence in the language.

Source: https://careers.google.com/stories/how-one-team-turned-the-dream-of-speech-recognition-into-a-reality/



[Left] A traditional monolingual speech recognizer comprising of Acoustic, Pronunciation and Language Models for each anguage. [Middle] A traditional multilingual speech recognizer where the Acoustic and Pronunciation model is multilingual, while the Language model is language-specific. [Right] An E2E multilingual speech recognizer where the Acoustic, Pronunciation and Language Model is combined into a single multilingual model.

Source: https://ai.googleblog.com/2019/09/large-scale-multilingual-speech.html

66. ZTE has had knowledge of the ’737 Patent at least as of the date when it was notified of the filing of this action.

67. Buffalo Patents has been damaged as a result of the infringing conduct by ZTE alleged above. Thus, ZTE is liable to Buffalo Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

68. Buffalo Patents has neither made nor sold unmarked articles that practice the ’737 Patent, and is entitled to collect pre-filing damages for the full period allowed by law for infringement of the ’737 Patent.

ADDITIONAL ALLEGATIONS REGARDING INFRINGEMENT AND PERSONAL JURISDICTION

69. ZTE has also indirectly infringed the ’086 Patent and the ’737 Patent by inducing others to directly infringe the ’086 Patent and the ’737 Patent.

70. ZTE has induced the end users and/or ZTE's customers to directly infringe (literally and/or under the doctrine of equivalents) the '086 Patent and the '737 Patent by using the accused products.

71. ZTE took active steps, directly and/or through contractual relationships with others, with the specific intent to cause them to use the accused products in a manner that infringes one or more claims of the patents-in-suit, including, for example, Claim 1 of the '086 Patent and Claim 13 of the '737 Patent.

72. Such steps by ZTE included, among other things, advising or directing customers, end users, and others (including third party testing and certification organizations) to use the accused products in an infringing manner; advertising and promoting the use of the accused products in an infringing manner; and/or distributing instructions that guide users to use the accused products in an infringing manner.

73. ZTE performed these steps, which constitute joint and/or induced infringement, with the knowledge of the '086 Patent and the '737 Patent and with the knowledge that the induced acts constitute infringement.

74. ZTE was and is aware that the normal and customary use of the accused products by ZTE's customers would infringe the '086 Patent and the '737 Patent. ZTE's inducement is ongoing.

75. ZTE has also induced its affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on its or its affiliates' behalf, to directly infringe (literally and/or under the doctrine of equivalents) the '086 Patent and the '737 Patent by importing, selling or offering to sell the accused products.

76. ZTE has a significant role in placing the accused products in the stream of commerce with the expectation and knowledge that they will be purchased by consumers in Texas and elsewhere in the United States.

77. ZTE purposefully directs or controls the making of accused products and their shipment to the United States, using established distribution channels, for sale in Texas and elsewhere within the United States.

78. ZTE purposefully directs or controls the sale of the accused products into established United States distribution channels, including sales to nationwide retailers. ZTE's established United States distribution channels include one or more United States based affiliates (e.g., ZTE (USA) Inc.).

79. ZTE purposefully directs or controls the sale of the accused products online and in nationwide retailers such as Amazon, including for sale in Texas and elsewhere in the United States, and expects and intends that the accused products will be so sold.

80. ZTE purposefully places the accused products—whether by itself or through subsidiaries, affiliates, or third parties—into an international supply chain, knowing that the accused products will be sold in the United States, including Texas. Therefore, ZTE also facilitates the sale of the accused products in Texas.

81. ZTE took active steps, directly and/or through contractual relationships with others, with the specific intent to cause such persons to import, sell, or offer to sell the accused products in a manner that infringes one or more claims of the '086 Patent and the '737 Patent.

82. Such steps by ZTE included, among other things, making or selling the accused products outside of the United States for importation into or sale in the United States, or knowing that such importation or sale would occur; and directing, facilitating, or influencing its affiliates,

or third-party manufacturers, shippers, distributors, retailers, or other persons acting on its or its affiliates' behalf, to import, sell, or offer to sell the accused products in an infringing manner.

83. ZTE performed these steps, which constitute induced infringement, with the knowledge of the '086 Patent and the '737 Patent, and with the knowledge that the induced acts would constitute infringement.

84. ZTE performed such steps in order to profit from the eventual sale of the accused products in the United States.

85. ZTE's inducement is ongoing.

86. ZTE has also indirectly infringed by contributing to the infringement of the '086 Patent and the '737 Patent. ZTE has contributed to the direct infringement of the '086 Patent and the '737 Patent by the end user of the accused products.

87. The accused products have special features that are specially designed to be used in an infringing way and that have no substantial uses other than ones that infringe the '086 Patent and the '737 Patent, including, for example, Claim 1 of the '086 Patent and Claim 13 of the '737 Patent.

88. The special features include, for example, display devices that have a front panel including a rigid substrate, a back panel including a flexible substrate, and a light control material in between the front and back panels, used in a manner that infringes the '086 Patent; and hardware and software components implementing a virtual keyboard interface that is capable of receiving freehand input using a stylus and audio input from a user, and capable of generating text data using multiple language models based, in part, on an adjusted language model, used in a manner that infringes the '737 Patent.

89. These special features constitute a material part of the invention of one or more of the claims of the ’086 Patent and the ’737 Patent, and are not staple articles of commerce suitable for substantial non-infringing use.

90. ZTE’s contributory infringement is ongoing.

91. ZTE has had actual knowledge of the ’086 Patent at least as early as November 26, 2021, when ZTE received a letter notifying it of the ’086 Patent, and/or as of the date when it was notified of the filing of this action. Since at least that time, ZTE has known the scope of the claims of the ’086 Patent; the products that practice the ’086 Patent; and that Buffalo Patents is the owner of the ’086 Patent.

92. ZTE has had actual knowledge of the ’737 Patent at least as of the date when it was notified of the filing of this action. Since at least that time, ZTE has known the scope of the claims of the ’737 Patent, the products that practice the ’737 Patent, and that Buffalo Patents is the owner of the ’737 Patent.

93. By the time of trial, ZTE will have known and intended (since receiving such notice) that its continued actions would infringe and actively induce and contribute to the infringement of one or more claims of the ’086 Patent and the ’737 Patent.

94. Furthermore, ZTE has a policy or practice of not reviewing the patents of others (including instructing its employees to not review the patents of others), and thus has been willfully blind of Buffalo Patents’ patent rights. *See, e.g.*, M. Lemley, “Ignoring Patents,” 2008 Mich. St. L. Rev. 19 (2008).

95. ZTE’s customers have infringed the ’086 Patent and the ’737 Patent. ZTE encouraged its customers’ infringement.

96. ZTE’s direct and indirect infringement of the ’086 Patent and the ’737 Patent, and its direct infringement of the ’405 Patent has been, and/or continues to be willful, intentional, deliberate, and/or in conscious disregard of Buffalo Patents’ rights under the patents-in-suit.

97. Buffalo Patents has been damaged as a result of ZTE’s infringing conduct alleged above. Thus, ZTE is liable to Buffalo Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

JURY DEMAND

Buffalo Patents hereby requests a trial by jury on all issues so triable by right.

PRAYER FOR RELIEF

Buffalo Patents requests that the Court find in its favor and against ZTE, and that the Court grant Buffalo Patents the following relief:

a. Judgment that one or more claims of the ’086 Patent, the ’737 Patent, and the ’405 Patent have been infringed, either literally and/or under the doctrine of equivalents, by ZTE and/or all others acting in concert therewith;

b. A permanent injunction enjoining ZTE and its officers, directors, agents, servants, affiliates, employees, divisions, branches, subsidiaries, parents, and all others acting in concert therewith from infringement of the ’737 Patent; or, in the alternative, an award of a reasonable ongoing royalty for future infringement of the ’737 Patent by such entities;

c. Judgment that ZTE account for and pay to Buffalo Patents all damages to and costs incurred by Buffalo Patents because of ZTE’s infringing activities and other conduct complained of herein, including an award of all increased damages to which Buffalo Patents is entitled under 35 U.S.C. § 284;

d. That Buffalo Patents be granted pre-judgment and post-judgment interest on the damages caused by ZTE's infringing activities and other conduct complained of herein;

e. That this Court declare this an exceptional case and award Buffalo Patents its reasonable attorney's fees and costs in accordance with 35 U.S.C. § 285; and

f. That Buffalo Patents be granted such other and further relief as the Court may deem just and proper under the circumstances.

Dated: April 27, 2022

Respectfully submitted,

/s/ Zachariah S. Harrington
Matthew J. Antonelli
Texas Bar No. 24068432
matt@ahtlawfirm.com
Zachariah S. Harrington
Texas Bar No. 24057886
zac@ahtlawfirm.com
Larry D. Thompson, Jr.
Texas Bar No. 24051428
larry@ahtlawfirm.com
Christopher Ryan Pinckney
Texas Bar No. 24067819
ryan@ahtlawfirm.com
Rehan M. Safiullah
Texas Bar No. 24066017
rehan@ahtlawfirm.com

ANTONELLI, HARRINGTON
& THOMPSON LLP
4306 Yoakum Blvd., Ste. 450
Houston, TX 77006
(713) 581-3000

Attorneys for Buffalo Patents, LLC