KWickChat: A Multi-Turn Dialogue System for AAC Using Context-Aware Sentence Generation by Bag-of-Keywords

Junxiao Shen
University of Cambridge
UK
js2283@cam.ac.uk

Boyin Yang
University of Cambridge
UK
by266@cam.ac.uk

John Dudley
University of Cambridge
UK
jjd50@cam.ac.uk

Per Ola Kristensson
University of Cambridge
UK
pok21@cam.ac.uk

ABSTRACT

We present KWickChat (Keyword Quick Chat): a multi-turn augmentative and alternative communication (AAC) dialogue system for nonspeaking individuals with motor disabilities. The central objective of KWickChat is to reduce the communication gap between nonspeaking and speaking partners by exploring a sentence-based text entry system that automatically generates suitable sentences for the nonspeaking partner based on keyword entry. The system is underpinned by a GPT-2 language model and leverages context information, including dialogue history and persona tags, to improve the quality of the generated responses. We evaluate the system by analyzing the functional design and decomposing it into key functions and parameters that are systematically investigated using envelope analysis. We pursue this methodology as a necessary precursor to evaluation with AAC users. Our results show that with word prediction and with a threshold word error rate of 0.65, the keystroke savings of the KWickChat system is around 71%. To complement the envelope analysis, we also recruited two human judges to evaluate the semantic consistency between 400 sentences generated by KWickChat and reference sentences. Both judges reported a median rating of 4 on a scale from 1 (very bad) to 5 (very good) for the best generated sentence in each exchange and achieved an inter-rater reliability of 0.92 across all 400 sentences judged.

CCS CONCEPTS

• **Human-centered computing** → **Natural language interfaces**; **Accessibility systems and tools**.

KEYWORDS

Natural Language Processing (NLP), Augmentative and Alternative Communication (AAC), Dialogue System, Deep Learning

IUI '22, March 22–25, 2022, Helsinki, Finland

ACM ISBN 978-1-4503-9144-3/22/03.
https://doi.org/10.1145/3490099.3511145

ACM Reference Format:
Junxiao Shen, Boyin Yang, John Dudley, and Per Ola Kristensson. 2022. KWickChat: A Multi-Turn Dialogue System for AAC Using Context-Aware Sentence Generation by Bag-of-Keywords. In *27th International Conference on Intelligent User Interfaces (IUI '22), March 22–25, 2022, Helsinki, Finland.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3490099.3511145

1 INTRODUCTION

Nonspeaking individuals with motor disabilities, such as amyotrophic lateral sclerosis (ALS) or cerebral palsy (CP), have lifelong conditions that affect movement, muscle control, and co-ordination. For these individuals with additional needs that make it difficult for them to use a keyboard or touchscreen, Augmentative and Alternative Communication (AAC) devices serve as tools to communicate with their speaking partners. AAC devices typically consist of either a physical keyboard, onscreen touch keyboard, or an onscreen keyboard that can be operated in different ways, such as via a switch system or using eye-tracking technology. An AAC device uses speech synthesis to transform text into natural human-like speech, which allows the user to engage in conversation with speaking partners. One critical challenge in AAC is the difference between the entry rates of AAC users (1–25 words per minute depending on the nature of the disability and the AAC device) and the entry rates of speaking users (150–200 words per minute) [5]. Such a difference in entry rates leads to a substantial *communication gap* between AAC users and their speaking partners.

Word prediction systems and context-aware sentence retrieval systems have been shown as two key systems that many AAC users rely on to a large extent (e.g. [15]). Such AAC devices are able to predict sentences customized for specific individual factors, such as the nature of the disability and corresponding needs, age, literacy level, occupation, personal preference, and other context information such as location and time. Prior research has demonstrated that context signals can boost performance and engage end-users on context sensing and its implications. Kristensson et al. [15] has demonstrated that such context-aware sentence prediction can, in theory, result in keystroke savings in the range of 50–96% depending on parameter assumptions. However, sentence retrieval systems have several drawbacks:

(1) Sentence retrieval systems only attend to the user's input and context information, but not to the speaking partner's content, not even to the previous turns of the conversation.

For example, question-focused retrieval systems reply to user's questions with pre-stored answers [26].

(2) Sentence retrieval systems typically ask a user to start typing from the beginning of a sentence [15]. Even though some studies use a keyword-based sentence-construction approach, they either require a well-defined semantic hierarchy in the dataset [17], or merely depend on a dataset of a limited size containing sentences with keywords. The strategy of typing a sentence from the beginning leads to the typing *stop words*, such as "a", "the", "is" and "are". These stop words are the most frequently used words in language and as such do not add any additional value to the sentence retrieval system.

(3) Information retrieval algorithms only perform well if the predicted sentence is among the stored sentences. The implication is therefore that a sentence retrieval-based AAC system only works reasonably well in constrained situations, such as daily routine conversation.

We propose KWickChat (Keyword Quick Chat) as a step to overcome these and other limitations. KWickChat is a multi-turn dialogue system that uses a language model known as Generative Pre-trained Transformer 2 (GPT-2), with bag-of-keywords and context information in order to generate high-quality sentence responses. The KWickChat dialogue system can intelligently predict and generate customized sentences for AAC users and therefore has the potential to server as a critical subcomponent within a comprehensive AAC system.

The system only requires a small number of keywords from the AAC user to generate a set of relevant and context-aware responses with high quality, so that the communication rates can be dramatically increased. Table 1 illustrates two conversations supported by the KWickChat system with four potential replies generated for each exchange. Each example is coupled with persona information that captures various details about the AAC user.

However, while promising, such a multi-turn AAC dialogue system is extremely difficult to validate with AAC users. This difficulty stems from both ethical issues associated with evaluating an incomplete system, as well as the requirement for extensive longitudinal observation to observe behavior change and benefits. Prior work has explored these issues in detail (e.g. [15]). At this stage in the design process, we therefore opt to perform extensive envelope analysis to explore the performance potential of the KWickChat system. We also conduct a human judgment analysis to examine the semantic similarity between predicted responses and target responses. Our focus in this paper is thus on evaluating the efficacy of the KWickChat dialogue system as a critical subcomponent of a comprehensive AAC system. Given this focus, we refrain from making any claims on the suitability of generated responses or specific interaction approaches.

An intelligent multi-turn dialogue system typically generates responses conditioned on the conversation history and the conversation partner. The novel contribution offered by this paper is to smoothly inject ideas from intelligent dialogue systems into AAC systems by generating utterances additionally conditioned on the user's input (bag-of-keywords), the conversation history and the user's persona.

In summary, the four key contributions of this paper are the following:

(1) We use a novel integration tactic for the transformer-based language models' capability to both summarize and create dialogue. The summary power is leveraged from Bidirectional Encoder Representations from Transformers (BERT) to extract keywords, while the creative power is leveraged from GPT-2 to generate realistic responses that are conditioned on bag-of-keywords, conversation history and persona.

(2) We bestow the AAC system with a multi-turn dialogue system that also utilizes persona details for additional contextual information in order to improve the quality of responses in a conversation.

(3) We introduce bag-of-keywords to efficiently reduce the user's cognitive load and improve entry rates by saving keystrokes. Using bag-of-keywords also bootstraps the language generation model to produce meaningful responses.

(4) We release our pre-trained KWickChat language generation model trained on 11,000 multi-turn dialogue samples from the ChatAI dataset [8]. This allows AAC designers and researchers to quickly deploy the model into an AAC system without the need to train their own model, which is a non-trivial task. In addition, this also allows fine-tuning the model on private datasets.

2 RELATED WORK

AAC systems enable users to communicate at faster rates with special input methods that boost the key selection rate [9, 22, 38] and predictive text entry models such as information retrieval approaches [34, 40]. In our work, we focus on the latter. Todman et al. [35] describe how a pause duration under 5 s is considered typical in a 15-20 minute social conversation with a speaking partner. Using a letter-by-letter selection keyboard, however, results in long pause durations (39.6 s on average) due to the delay in typing. Hence, Todman et al. [36] suggested that using utterance-based devices (UBDs) would rapidly increase the communication rate. UBDs allow users to select the most relevant phrases or sentences from pre-stored conversation material by searching through the hierarchical storage structure instead of entering text letter-by-letter. However, by their nature UBDs have limitations in conducting new communication topics or handling new situations as such systems are unable to construct new sentences with the language material outside of the dataset [2]. By contrast, we propose using keywords to produce the utterance and a topic-free language generation model. This eliminates the low-entry rate issue of letter-by-letter methods and also avoids the conversation topic limits characteristic of UBDs.

Context information including time and location can reduce the language error rate in text entry models [29]. Data used for sentence retrieval can be strongly context correlated. For example, a reply is typically tightly related to the question. Different context tags such as semantic tags and geographical position tags can be added to allow faster and more precise retrieval of phrases [12, 28]. Therefore, we adopt context information as an essential input to the language generation model to increase prediction accuracy.

Natural language processing (NLP) has been leveraged in AAC systems to improve the efficiency of communication [6, 13, 17,

Persona:	i like to remodel homes.	i like to go hunting.	i like to shoot a bow.	my favorite holiday is halloween.
Exchange	Conversation Partner	Bag-of-Keywords	Generated Sentences (with selected sentence in bold)	
1	hi, how are you doing? i'm getting ready to do some cheetah chasing to stay in shape.	fast hunting hobbies	**1. i'm good. i like to go fast hunting. what are your hobbies?**	
			2. i like to go for a run. i also like to go for a run.	
			3. i like to go for long runs too. fast hunting is one of my hobbies.	
			4. what are some of your favorite hobbies?	
2	for my hobby i like to do canning or some whittling.	remodel bow hunting	1. i like to remodel homes. i like to remodel homes	
			2. me too! i remodel homes. i like to remodel homes.	
			3. i like to remodel. i do not have a bow and hunting bow.	
			4. i remodel homes but i do not have a bow and huntingbow.	

Persona:	i drive a 2015 honda civic.	my favourite food is pizza	i listen to rap music.	my pets name is charlie.
Exchange	Conversation Partner	Bag-of-Keywords	Generated Sentences	
1	hello, how are you this evening?	good who	**1. i'm good who are you?**	
			2. i am good. what are you up to?	
			3. i am good what are you up to?	
			4. i am good and you?	
2	i'm bryan the rose gardener, who are you?	philip charlie listening music	**1. the name is philip, i am listening to rap music right now with my pet charlie.**	
			2. i am philip. what are you listening to? i am listening to rap music.	
			3. i am philip, i am listening to rap music right now.	
			4. that is cool. i am philip, and i am listening to rap music.	
3	awesome, i am hanging out in my big in home library	nice what book	1. that's nice. i am listening to rap music right now.	
			2. that sounds nice. what book?	
			3. nice one. what book is it?	
			4. it is nice what is your name?	
4	a book on how to read lips, i am half deaf	interesting skill	**1. that's a very interesting skill?**	
			2. it is an interesting skill	
			3. yes it can be an interesting skill	
			4. it is an interesting skill	

Table 1: Two examples illustrating sentences generated by KWickChat in a conversation.

39]. Schadle [33] proposed letter prediction and word prediction methods using NLP techniques that dynamically place the next-key-to-press and allow keystroke savings by word completion. In NLP-based text prediction, words, phrases and sentences are encoded and stored for future retrieval [1]. These methods have been applied to AAC for the assistance of story-telling and social dialogue [31].

In an early study of generating sentence predictions, Copestake [5] combined word prediction, keyword extraction, and natural language generation methods to improve the typing rate. Copestake [5] observed that the co-generation approach developed produces useful predictions but no quantitative evaluation is presented. Some AAC researchers [14, 18] indicate that AAC systems should consider not only individual features such as keystroke savings and

sentence prediction, but also the conversation as a whole. A well-designed AAC system should enhance the overall communication rate. Therefore, we take the conversation history into account as an essential part of context information for sentence generation. With the assistance of modern neural network models [30], our work presents a multi-turn AAC dialogue system with a powerful sentence co-generation model on the basis of user-input keywords, conversation history and user persona.

AAC system design and evaluation in general is challenging, a fact which has been reflected on before (e.g., [15]). The various challenges encountered can short-circuit established user-centered design practices for several reasons [15]: i) a context-aware sentence prediction system needs to be bootstrapped for one specific user to flourish with the context information such as persona and dialogue history; and ii) AAC users have limited text entry rates which suggest that a thorough user-centered evaluation may require months of use for a proper evaluation. In addition, other alternatives, such as using proxy-users pretending to be nonspeaking individuals with motor disabilities, cannot reproduce accurate behavior of people with disabilities.

A recent HCI strategy that tackles this challenge is an approach adopted from design engineering [15, 16]. Briefly, the central idea is to arrive at a functional design of the system that decouples functions from function carriers (actual implementations). This functional design can then be parameterized into controllable and uncontrollable parameters which can subsequently be investigated in computational experiments to assess performance potential. Prior work has used this approach successfully for analyzing sentence prediction for AAC [15] and the efficacy of predictive text on mobile phones [16].

3 LANGUAGE MODELS

We use three models in the paper for sentence retrieval, sentence generation and bag-of-keywords extraction respectively. Later in Section 9 we use Term Frequency-Inverse Document Frequency (TF-IDF) [32] as a baseline sentence retrieval model. As previously mentioned, we introduce sentence generation based on bag-of-keywords, together with conversation history and persona. GPT-2 is used for sentence generation [30] while BERT [7] is used to extract the bag-of-keywords from the training dataset in order to train GPT-2 as the KWickChat language generation model. Both of these models belong to the transformer family. A transformer, at a high level, is a stack of encoders and decoders of the same number, with each encoder consisting of a feed-forward neural network and a self-attention layer, and each decoder having extra encoder-decoder attention. Attention is used to improve the performance in transforming sparse indices to contextual embeddings. Self-attention focuses on relating different positions of the input sequences within each other, whereas encoder-decoder attention is used to tackle the challenge of alignment between the layer inputs and outputs within the decoder. GPT-2 is built using transformer decoder blocks while BERT is built using transformer encoder blocks.

3.1 KWickChat Language Generation Model

We apply GPT-2 to improve the quality of the response generation. GPT-2 is a state-of-the-art transformer-based language model for sentence generation. GPT-3 and other advanced transformer-based language models are also available [3], but these newer models are either not yet publicly available or are too big for non-enterprise use. GPT-2 is pretrained on a massive 40GB dataset called WebText, allowing it to generate long stretches of contiguous coherent text [30]. In simple terms it acts like the next word prediction feature of a keyboard application as it only outputs one token (or word) at a time. GPT-2 is an improved version of GPT-1 and contains more layers and is trained on a larger training dataset. GPT-2 (1.5B parameters) has ten times as many parameters as GPT-1 (117M parameters).

The GPT family is built based on the Transformer, which is an unsupervised attention-based model composed of an encoder and a decoder. Both the encoder and decoder consist of many feed-forward and attention layers stacked on top of each other [37]. Attention layers strengthen the model in attending to the critical part of the input sequence. We desire such attention power so that keywords and other important contextual information can be attended and embedded. Prior to Transformer, most natural language processing models were supervised models and needed to be trained for specific tasks, such as textual entailment and sentiment classification. These supervised models have major limitations, including the requirement for a large amount of task-specific annotated data and poor ability to generalize to unobserved data. Transformer addresses these limitations by being a generative model that is trained on unlabeled data and then enabling domain-specific researchers to fine-tune the model for a specialized downstream task. GPT-2 is a decoder-only Transformer in that it takes input by embedding the input word into its embedding matrix which incorporates positional encodings and token embeddings. Positional encodings are used to indicate the order of the words in the sequence and also the sentiment segments, if necessary. Thereafter the embedding matrix is passed through many layers of decoders and each decoder is a stack of self-attention layers and feed-forward layers. Self-attention is critical here as it allows the model to understand the relevant and associated words that explain the context of a certain word before passing it to the feed-forward layer. For example, "an AAC system should follow its ..." could be an input to GPT-2, and the self-attention layer will thus pay attention to "an ACC system" when it processes the word "its". Such attention is calculated as attention scores, which are used to generate tokens.

We adapt GPT-2 to provide KWickChat's language generation model and the integrated system delivers five central features:

(1) **Context-aware system:** KWickChat leverages what we call persona details that capture personal information about the specific user. These details are encoded as sentences, such as "I am a researcher interested in accessibility." This provides a special type of context information. In Table 1, we see in exchange two of the first conversation that by only entering "remodel", the generated sentence contains "remodel homes" because this background information is included in the persona details.
(2) **Multi-turn system:** Multi-turn dialogue systems use different language generation models to predict high-quality responses conditioned on conversation history [24, 45]. The conversation history is defined as the previous exchanges

within the current conversation. We enabled this functionality by concatenating the conversation history into the input of the language generation model so that the model can generate dialogue conditioned on the conversation history. We measure the size of history by the number of exchanges in the conversation.

(3) **Bag-of-keywords to save keystrokes:** We allow the user to input keywords to substantially save keystrokes and boost entry rates by avoiding the need to enter stop words. We propose bag-of-keywords which models the keywords entered by the user during the inference phase and the keywords extracted from training dialogue data during the training phase. A bag-of-keywords is a multiset of terms that, unlike a set, preserves the multiplicity of the terms. The use of a bag-of-keywords can also help the language model understand and attend to the key contextual information of a sentence during training and inference. Table 1 illustrates the importance of entering keywords to enable the generation of high quality responses. Entering more keywords helps to constrain the variation in the sentences generated while entering fewer keywords generally results in more varied sentences.

(4) **Open-domain AAC model:** The KWickChat language generation model can generate meaningful novel sentences conditioned on the content of the conversation. It is open-domain in that it is not specifically trained for one particular context of use. In contrast, sentence retrieval-based AAC systems leveraging a keyword-based sentence-making strategy cannot generate unobserved sentences and thus can only work in certain contexts, such as daily routine conversations. We leveraged GPT-2 and trained it on an open-domain dataset consisting of highly variable content, leading to a robust sentence generation model. We also foresee that if the model is trained on a dataset from a specific user, it can learn the user's preferences and conversation behaviors and embed them into the neuron units to generate text representative of their personal "voice". Table 1 demonstrates that the conversation is not constrained to a specific domain and users can freely converse on different topics.

(5) **The size and memory consumption of a deep-learning based model remain constant:** GPT-2 is a deep neural network model that learns the representation of a generative function by updating the parameters of the model through back-propagation. Therefore, the model's size stays constant regardless of how much data is used to train the model, making the model a memory-efficient approach to 'store' previous information. Generally, when a deep-learning-based model is trained on more data it becomes better at extracting latent attributes of the data and approximating the generative function.

The following subsections introduce how GPT-2 is adapted for our task of generating high-quality responses conditioned on three context vectors: persona, history and bag-of-keywords.

3.1.1 Adding Context to GPT-2. This subsection focuses on the following challenge: how can we combine the three context sources as input to the model while at the same time also allow the model to distinguish between these sources?

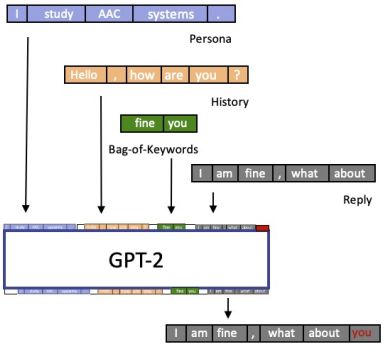


Figure 1: The four context segments are concatenated and used as the input to the GPT-2. Another input is the segment embeddings which are represented by the different colors. The predicted tokens, indicated by the red segment and red text in the figure, are used as the last segments in the concatenated segments to enable token-by-token prediction. Adapted from Wolf [42].

We have the following context sources:

(1) one or more persona (background information about the user);
(2) one or more exchanges in the conversation history; and
(3) bag-of-keywords entered by the user.

Given that GPT-2 generates the output sequence word by word in a sequential fashion we also have an additional word embedding representing the collection of previously generated tokens. We first concatenate the context segments into a single sequence. Then we produce segmented embeddings by adding: 1) special tokens for delimiters (including start-of-sentence and end-of-sentence symbols); 2) segment indicators; and 3) positional information for each token. The words and segments embeddings are represented by the different colors in Figure 1. We then iteratively generate a complete sentence one token at a time.

3.1.2 Multi Task Heads. There are two challenges in generating high-quality responses: 1) the generated responses must make sense from both a grammatical and semantic perspective; and 2) the generated responses must reflect the context input. We tackle this by using two heads connected to the model [43]: the language modeling head addresses the first challenge while the word/sentence prediction head addresses the second challenge. A multi-task loss is used which consists of:

- A loss from language modeling: we project the hidden-state on the word embedding matrix to obtain output logits, which take the form of a vector of non-normalized probability predictions for each character, and apply a cross-entropy loss on the portion of the target corresponding to the golden reply, which is the true reply.
- A loss from next-sentence prediction: we pass the hidden-state of the end-of-sentence token through a linear layer to

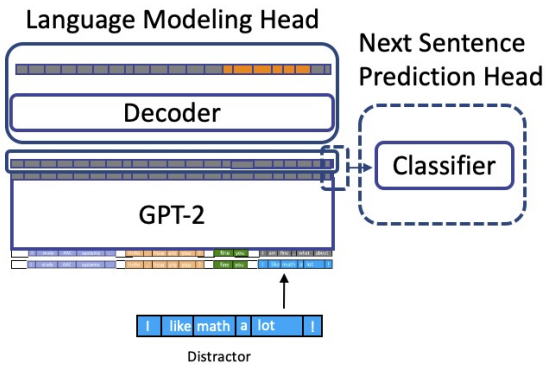


Figure 2: Two model heads—one for language modeling to generate responses with high quality in grammatical and semantical aspects, and one for next-sentence prediction to ensure the match between the generated responses and the context input. Adapted from Wolf [42].

obtain a score and apply a cross-entropy loss to classify a true reply among distractors.

Figure 2 shows the multiple heads and their corresponding tasks.

3.1.3 Decoding. The final output logits are decoded using top-p (nucleus) sampling. Top-p sampling truncates the tail of a probability distribution, that is, it retains only a subset S of the candidates, where S is the smallest subset whose total probability mass is greater than or equal to the threshold top-p. The method then samples from the dynamic nucleus (subset) containing the majority of the probability mass.

3.2 Bag-of-Keywords Extraction

There are currently no dialogue datasets that naturally come with keywords as a data feature. Therefore, we use BERT to extract keywords from the dialogue to prepare a training dataset and a validation dataset for the sentence generation model. BERT, unlike GPT-2, which is auto-regressive in nature, loses auto-regression but can incorporate context on both sides of a word to gain better results on certain tasks, such as keyword extraction—owing to its bi-directional transformer structure [7]. We first use BERT to convert the user's intended sentence to high-quality embeddings so that the meaning of the sentence is captured [11]. Then the word embedding is extracted for n-gram words from the sentence embedding. We then use cosine similarity to measure the difference between the word embedding vectors and the intended sentence embedding so that the most similar words can be identified to be the keywords that best describe the intended sentence. Figure 3 summarizes the overall training and inference process for the proposed KWickChat language generation model, including the role BERT plays in extracting keywords.

4 DATASET

We conduct experiments on the ConvAI2 challenge dataset which is an altered version of the PersonaChat dataset [8, 44]. The dataset consists of conversations between crowd workers who were randomly paired and asked to act the part of a given person (randomly assigned from 1,155 possible persona that were created by another set of workers), chat naturally, and get to know each other during the conversation. The persona represents the personality context tags of the speaking person. There are around 160,000 utterances in around 11,000 dialogues, with 2,000 dialogues for validation and testing. Table 2 illustrates one example from the dataset. In the AAC setting, person 1 can be viewed as the speaking partner, while person 2 can be viewed as the AAC user. The bag-of-keywords are extracted from the replies of person 2 to form a training dataset to train the language generation model.

The ConvAI2 dataset is commonly used for training and validating a multi-turn chatbot. Several other factors also motivate its use in training KWickChat's language generation model. First, there is no existing conversational dataset produced by AAC users. This stems from the fact that such a dataset would be very difficult to collect due to typically slow typing rates, significant diversity in AAC systems, as well as privacy issues associated with collecting data from people who rely on these devices as their sole communication device [15]. Second, this paper focuses on developing a language generation model based on bag-of-keywords input. Training such a language generation model leveraging different context information and bag-of-keywords requires a very large labeled dataset. Given the difficulties associated with collecting a very large AAC specific dataset, it makes sense to leverage existing datasets to produce a pre-trained model. This trained generative model can then be further fine-tuned on a more AAC user-focused dataset—if and when such a dataset becomes available. Theoretically, the model could also be trained with data from one specific user such that the model can adapt to reflect the user's own preferences and behaviors. Given this lack of an existing AAC dataset, and the need for a large-scale conversation-focused dataset, we selected the ConvAI2 challenge dataset to train the language generation model. The ConvAI2 dataset also has several other qualities that motivate this choice:

(1) It fits well with our use case, given that it contains different context information, such as personas.
(2) It is well-structured with multiple exchanges in each conversation, making it possible to perform the parameter analysis presented in this paper.
(3) It contains distractor replies in each conversation that can be randomly sampled to be compared with the golden reply to construct a cross-entropy loss to train the language modeling head.

5 IMPLEMENTATION DETAILS

We used a computer that has 3 × Nvidia GeForce RTX 3090 GPU and 1 × Ryzen 9 3970x CPU. We built our model based on the conversational artificial intelligence source code implemented by Wolf et al. [43]. We extract bag-of-keywords of sizes randomly chosen between 1 and the maximum number of keywords in the training sentence, and combine this information with the history and persona to form the input to the GPT-2 language generation model during training. We use the pre-trained GPT-2 from OpenAI [25, 30]. We set the maximum number of previous exchanges to

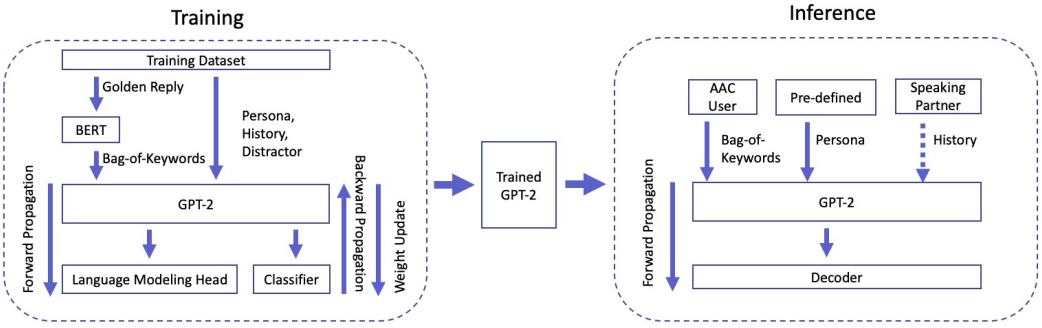


Figure 3: The overall process flow of the proposed KWickChat language generation model. Firstly, a GPT-2 is trained/fine-tuned using the ConvAI2 challenge dataset as the training dataset. The trained/fine-tuned GPT-2 model is then used to perform inference from the inputs including bag-of-keywords, persona and history to generate high-quality responses. Note that the history does not only come from the speaking partner but also from the system itself, which are the selected responses by the AAC user.

Persona 1	Persona 2
I like to ski	I am an artist
My wife does not like me anymore	I have four children
I have went to Mexico 4 times this year	I recently got a cat
I hate Mexican food	I enjoy walking for exercise
I like to eat cheetos	I love watching Games of Thrones
PERSON 1:	Hi
PERSON 2:	Hello ! How are you today ?
PERSON 1:	I am good thank you, how are you
PERSON 2:	Great, thanks! My children and I were just about to watch Games of Thrones.
PERSON 1:	Nice ! How old are your children?
PERSON 2:	I have four that range in age from 10 to 21. You?
PERSON 1:	I do not have children at the moment.
PERSON 2:	That just means you get to keep all the popcorn for yourself.
PERSON 1:	And Cheetos at the moment!
PERSON 2:	Good choice. Do you watch Game of Thrones?
PERSON 1:	No, I do not have much time for TV.
PERSON 2:	I usually spend my time painting: but, I love the show.

Table 2: Example multi-turn dialogue from the ConvAI2/PersonaChat challenge dataset, adapted from [8]. Person 1 and 2 are given their own persona (top) at the beginning of the chat but do not know each other's persona. Participants were instructed to get to know each other during the conversation. In this example, there are 6 exchanges in the conversation. The number of persona details for Person 1 is 5.

keep in history to be 3. The batch size for training and validation are both 4. The gradients are accumulated every 8 steps. The model is trained for 3 epochs. The batch size, gradient steps and the number of epochs are chosen to fit the maximum memory of the available GPU. The loss coefficient for the language modeling head is 1.0 and the loss coefficient for the next sentence prediction head is 1.0. The learning rate of training is set to be 6.25e-5. We also clip the gradient to be within 1 to prevent gradient explosion. The rest of the values are adopted from the original implementation [43].

6 QUANTITATIVE PARAMETER EXPLORATION

In this section, we apply a methodology closely related to the modeling and envelope analysis approach proposed by Kristensson et al. [15]. This methodology is motivated by the significant ethical issues associated with evaluating an untested prototype AAC system with AAC users. AAC users rely on their AAC system for all communication and it is inappropriate to burden such users and limit their communication ability without first proving the functionality of the prototype system in other ways. Therefore, in this paper, we first seek to validate the system through analytical means and this is the underlying rationale for the design of the system evaluation.

First, we model the sentence generation problem as an information generation (IG) problem, which is a modified version of an information retrieval (IR) problem because we generate information instead of retrieving information. We assume the AAC system has deployed the trained GPT-2 as the language generation model and that the model is trained on the ConvAI2 challenge dataset. The ConvAI2 challenge dataset is a multi-turn dialogue dataset that consists of golden replies from participants, their corresponding persona and the conversation history. The term *golden reply* is a widely used term in dialogue system nomenclature [19] and it describes the actual sentence entered by the user.

We model different users by assigning them various persona and the length of the conversation is modeled by setting the number of exchanges in the conversation history. Each golden reply is associated with one or more persona and one or more exchanges in the history. The number of exchanges in the conversation history indicates the number of back-and-forth turns in a conversation, therefore, the number of exchanges is by definition the number of golden replies in the conversation minus one. Both the persona and the history are treated as terms. We also include the bag-of-keywords extracted from a golden reply as the term and the collection of the terms form a *document*. Here, a bag-of-keywords is slightly different from a bag-of-words in IR nomenclature. A bag-of-words models the whole *document* and also includes the sentence (golden reply in our case) [15], whereas a bag-of-keywords only models the golden reply. During training, each bag-of-keywords is extracted from a golden reply by BERT. This *document* is fed into the language generation model to output logits which will then be decoded into a *predicted reply*. This reply will be compared with the *golden reply* to evaluate the loss function so that the language generation model can be trained.

A *query* consists of a bag-of-keywords the user has entered, the persona of the user and the history (if available). This *query* will be fed into the language generation model to output a *predicted reply*. In developing KWickChat, we have assumed that we do not know the length of the *golden reply* given that when a hypothetical user starts to write a sentence, they may have a vague idea of the representation of the sentence but do not know the exact length of the intended sentence until a *predicted reply* matches their intention.

Word prediction (auto-complete assistance) is a Boolean parameter (supported or unsupported) in modeling the user's typing. A user needs to type each individual keyword until completion without aid if not using word prediction. Once a keyword is completed, it is added to the bag-of-keywords and the updated bag-of-keywords is used to update the *query*. Then the *query* is formed for every keyword typed regardless of the typing method, together with any present bag-of-keywords, persona and history. The *query* is then passed to the language generation model for prediction. The output logits from the generation model are decoded using top-p (nucleus) sampling. As the sampling strategy can produce different responses by altering the random seed, the seed is changed multiple times so that several sentences can be generated and displayed for the user to select. Each selection also represents one keystroke. This models an interactive approach where, every time the user finishes entry of a keyword, KWickChat will output several predicted sentences for display, which enables the user to choose from a variety of responses. We consider a match if one of the displayed sentences matches the *golden reply* by passing a criterion threshold. This typing process repeats until the system is able to generate a matching sentence. We then calculate the keystroke savings, a metric that is defined in Section 6.1.2.

6.1 Evaluation Metrics

6.1.1 Function-Level Evaluation. Ranking or retrieval models are typically evaluated based on whether they can retrieve the correct responses, which includes the ground truth response to the conversation. Such systems can be evaluated using recall or precision metrics. However, when deployed in a real setting these models will not have access to the correct response given an unseen conversation. Further, a sentence prediction model may not predict the exact same sentence as the target sentence but may still capture the same meaning.

It is challenging to find a high-quality evaluation metric to evaluate the performance of the models. We have different models to evaluate, therefore, we only consider metrics that are model-independent, that is, where the model generating the response does not also evaluate its quality. Therefore, perplexity is not considered as it is not computed on a per-response basis and cannot be computed for retrieval models. There are many different model-independent metrics to evaluate the performance of such a sentence prediction model, mainly divided into word overlap-based metrics and embedding-based metrics. We use the following metrics to perform model evaluation and also compare the correlation between the different metrics.

The word overlap-based metrics, such as Bilingual Evaluation Understudy Score (BLEU), evaluate the amount of word-overlap between the proposed response and the ground-truth response. We are aware that recent studies suggest that many metrics commonly used in the literature, such as BLEU, do not correlate strongly with human judgment in evaluations of unsupervised dialogue systems [20, 21]. These prior analyses focus on relatively unconstrained domains. However, Liu et al. [20] also point out that for applications in constrained domains, there may be stronger correlations with the BLEU metric. For example, Wen et al. [41] propose a model for natural language generation on spoken dialogue systems and use BLEU to evaluate the quality of the generated sentences. Our model is conditioned on bag-of-keywords and thus provides a constrained domain for dialogue generation. Therefore, we suggest that BLEU is an appropriate metric for evaluating the quality of the responses. However, given the recognized issues with BLEU, we also include a variety of different model-independent metrics, and we demonstrate that they are correlated in Section 7. Word Error Rate (WER) is another word overlap-based metric which has been shown to correlate with human judgment in target domains.

(1) **BLEU**: BLEU has frequently been reported as correlating well with human judgment [27]. BLEU is an algorithm for evaluating the quality of text generated from a language model by analyzing the co-occurrence of n-grams in the ground truth and the proposed responses. BLEU ranges from 0 to 1: closer to 1 represents more similarity between the candidate text and the reference text. It is not necessary to obtain a BLEU score of 1 as this would indicate that the two texts are identical. If there is no n-gram overlap for any

order of n-gram, BLEU returns 0 because the precision for the order of n-gram without overlap is 0 and the geometric mean in the final BLEU score computation multiplies the 0 with the precision of other n-grams. We use a smoothing function on a sentence level to avoid this behavior [4].

(2) **WER**: WER is the minimum number of word insertions, deletions and substitutions necessary to transform a source sentence into a target sentence, divided by the number of words in the target sentence.

The embedding-based metrics consider the meaning of each word as defined by a word embedding, which assigns a vector to each word. These embeddings are calculated using distributional semantics; that is, they approximate the meaning of a word by considering how often it co-occurs with other words in the corpus [19]. The embeddings can be extracted by methods such as Word2Vec [23]. We use Word2Vec vectors trained on the Google News Corpus as the word embeddings [10]. The embedding-based metrics include Greedy Matching, Vector Average and Vector Extrema [20]:

(1) **Vector Average**: Vector Average is defined as the cosine similarity of the average scores of all word vectors composing a source sentence and a target sentence.

(2) **Greedy Matching**: Greedy Matching tries to find the maximum cosine similarity on a word-to-word basis, where each word of the source sentence is matched against all words of the target sentence to find the maximum cosine similarity. Then these maximum cosine similarities for all words in a source sentence are summed up and normalized by the length of the source.

(3) **Vector Extrema**: For each dimension of the word vectors, the extrema value is selected using the maximum value of the absolute value of the minimum and maximum of the corresponding dimension. The sentence vector is then created out of these extrema values from all dimensions. A final score is computed by taking the cosine similarity of the source sentence vector and target sentence vector.

6.1.2 System-Level Evaluation. The KWickChat system consists of the GPT-2 based language generation model as well as other modules for displaying suggested sentences and providing word predictions. We use keystroke savings (KS) (e.g. [15]) to quantify the performance of KWickChat as a system, where:

$$\text{KS} = (1 - \frac{k_m}{k_c}) \times 100\%. \quad (1)$$

Here k_m is the number of keystrokes that need to be typed before the model under investigation results in a matching sentence and k_c is the number of keystrokes in total for the test sentence. A higher keystroke savings value suggests an AAC system with potentially higher performance. Kristensson et al. [15] envisaged a sentence retrieval system to predict unobserved sentences that are not stored in the system. Therefore, by definition, such unobserved sentences are unlikely to perfectly match stored sentences. We face a similar problem in that the sentence generated is unlikely to exactly match the golden reply. For example, a target (unobserved) sentence of, "I want to eat a burger" and a predicted sentence, "Can I eat a burger" may be considered a good prediction result by the user even though they are different responses. Therefore, following a similar strategy as proposed by Kristensson et al. [15], we use WER to determine whether or not a predicted sentence is considered to be correct. A threshold is set on WER and then we observe the value of average keystroke savings by varying the threshold WER. If the threshold is passed, the predicted sentence becomes the test sentence with k_c keystrokes and the number of keystrokes for entering the keywords including the space key is k_m. If a golden reply is not predicted when the user enters all the keywords, that is, the predicted sentence did not pass the WER threshold, then the keystroke savings is 0%.

7 SURROGATE USER MODEL

We first investigate the effect of the size of the bag-of-keywords in the performance of the language generation model by creating a surrogate user model that can generate a bag-of-keywords from the intended (*golden reply*). We derive a surrogate user model by employing the generative model from BERT and introduce a parameter representing the number of the keywords extracted from the *golden reply*. To illustrate the intrinsic improvement brought by the bag-of-keywords, we do not concatenate the history and persona to the input of the language generation model and thereby temporarily eliminate the effect of these two pieces of context information.

Figure 4 and 5 plot the BLEU score and WER against the number of keywords entered by the surrogate user model. We also include the BLEU score and WER between the golden reply and the raw set of keywords to help highlight the intrinsic performance of the sentence generation model. The difference in BLEU score between the raw set of keywords and the generated sentence is much larger than the difference observed in the WER score. This is because WER measures the word-level similarity whereas BLEU looks for n-grams when measuring similarity. Keywords on their own do not typically form a sensible n-gram. Figure 4 and 5 show that performance increases as the number of keywords increases but saturates when the number of keywords is large. Figure 5 suggests that by only entering 4 to 5 keywords, a user can achieve a result of similar quality to that produced by a larger number of keywords. The observed impact of the number of keywords is expected given that more keywords transfer more information so that the model can make better predictions. Table 3 presents a specific example of how the increase in the number of keywords can improve the quality of the predicted sentence.

We use BLEU and WER to describe the performance of the model. The generated sentences were also evaluated using the embedding-based metrics previously introduced. We plot the value of each metric against the BLEU score for each operating point evaluated. We can see from Figure 6a, 6b and 6c that all the embedding-based metrics have a positive correlation with the BLEU score. Table 4 shows several generated responses and their corresponding metric scores. We see that the BLEU score can be as high as 0.89 in the table, but the average BLEU score in Figure 4 does not exceed 0.17. This is because the BLEU score is averaged over the validation data and some of the BLEU scores are approaching zero. Figure 6d suggests that WER has a negative correlation with BLEU, which is also expected and aligns well with our previous observations. Given that the various metrics have a clear correlation, we proceed in the following section to only use the BLEU score as the metric

Partner's Utterance:	**neat!! i used to work in the human services field**		
User Keywords	fun hard	neat stunt double fun	neat stunt double fun hard
Golden Reply	yes it is neat, i stunt double, it is so much fun and hard work.		
KWickChat	that sounds like fun. it is hard to get enough of it.	that is neat, i am a stunt double. what do you do for fun?	that is neat, i am a stunt double, its fun and hard.
BLEU score	4.78e-155	0.17	0.25
WER score	0.83	0.77	0.5
Vector Average score	0.9928	0.9908	0.9916
Vector Extrema score	0.92	0.81	0.80
Greedy Matching score	0.68	0.66	0.64

Table 3: Examples showing the benefits of a larger bag-of-keywords.

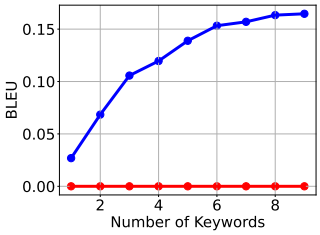


Figure 4: The blue line plots the average BLEU of the generated sentence with respect to the target sentence against the number of keywords input to the model. The red line plots the BLEU score from just entering the keywords without feeding them to the KWickChat language generation model to form a complete sentence. The more keywords the user enters, the better the quality of the generated response. The reference score from the keywords stays close to zero as keywords do not form a sentence, and BLEU uses n-gram modeling to measure the similarity with the generated response.

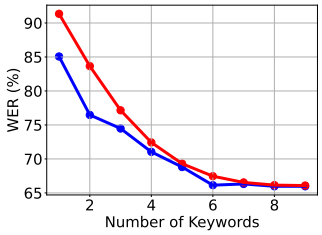


Figure 5: The blue line plots the average WER of generated sentence with respect to the target sentence against the number of keywords input to the model. The red line represents the WER from just entering the keywords without feeding them to the KWickChat language generation model to form a complete sentence. Just entering keywords approaches the generated sentences when the number of keywords is large, because WER only measures word-level similarity.

for evaluating the impact of contextual information on the sentence generation model.

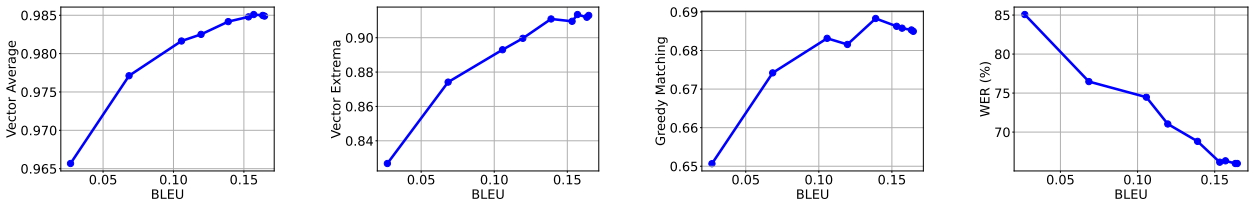
8 SURROGATE CONTEXT MODEL

We analyze the effect of the context, including persona and history, on the language generation model. We build a surrogate context model that can generate conversation sets with associated persona. Each conversation set has a number of exchanges in conversation history (size of history) and persona. As the user types during a conversation, an actual system would record or transcribe the previous exchanges in the same conversation and provide pre-set information on persona. The bag-of-keywords typed by the user and the history and the persona of the user together form a *query* fed into the language generation model in real-time. The history and persona, as illustrated in Figure 1, are concatenated to the start of the *query*. This means that part of the *query* is pre-loaded with terms before the user has begun typing. The controllable parameters of the surrogate context model are: i) the number of exchanges in the conversation history (default set to 2, unless otherwise stated); and ii) the number of persona (default set to 3, unless otherwise stated).

We study the effect of the size of persona and the size of exchanges in conversation history on the performance of the model and the system. Figure 7a and 7c plot keystroke savings and BLEU against the size of history. Figure 7a and 7c show that both the sentence generation model and the KWickChat system benefit from an increase of history size, which is the number of exchanges in the conversation history. Figure 7b and 7d also show that both the model and system have improved performance as the number of persona tags increase. This means users are likely to experience better results from the system in a conversation when the conversation is longer. Note that when computing keystroke savings a successful sentence is generated when the WER score between the source sentence and target sentence is within a threshold. We set the WER threshold to 0.65. Figure 5 shows the average WER value approached when a large number of keywords are provided. We assume this is the best performance the KWickChat language generation model can achieve. Therefore, by setting the threshold

Partner's Utterance:	**hi! do you like turtles?**	**i used to do home health aide but now i am disabled.**	**i went to school to be a vet, but i didn't like it**
User Keywords	cat person actually	sorry	want teacher grow
Golden Reply	i am much more of a cat person actually	i am sorry to hear that. what happened	i want to be a teacher when i grow up.
KWickChat	i'm more of a cat person actually	oh no. i am sorry to hear that.	i want to be a teacher when i grow up.
BLEU score	0.61	0.64	0.89
WER score	0.33	0.56	0.0
Vector Average score	0.9969	0.9745	1.0
Vector Extrema score	0.99	0.91	1.0
Greedy Matching score	0.66	0.61	0.67

Table 4: Examples showing the different metric values for different replies generated by KWickChat.



(a) Vector Average **(b) Vector Extrema** **(c) Greedy Matching** **(d) WER**

Figure 6: Correlation between BLEU and different metrics. The three embedding-based metrics all share a positive correlation with BLEU and a negative correlation with WER.

to 0.65, the selected responses produced by the KWickChat language generation model should be of a reasonable quality. It is also worth noting that the choice of this threshold does not affect our interpretation of the effect that the size of history and persona tags have on keystroke savings, provided the threshold is constant.

9 BASELINE COMPARISON

We evaluate the transformer-based sentence generation model against a baseline provided by TF-IDF. TF-IDF is a statistic to compute a weight for each word that signifies the importance of the word in the document and corpus. TF-IDF is calculated as the product of *Term Frequency (TF)* and *Inverse Document Frequency (IDF)*. *Term Frequency* measures the frequency of a word in a document. *Inverse Document Frequency* is used to measure the importance of a document in a whole document set (corpus) with count N. Document (d) here refer to a set of words (t).

$$\text{TF-IDF}(t,d) = \text{TF}(t,d)\log\left(\frac{N}{\text{DF}(t)+1}\right), \tag{2}$$

$$\text{where TF}(t,d) = \frac{\text{count of } t \text{ in } d}{\text{number of words in } d} \tag{3}$$

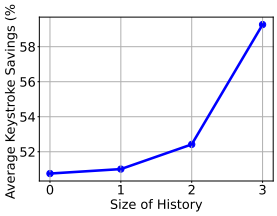
$$\text{and DF}(t) = \text{occurrence of } t \text{ in documents.} \tag{4}$$

We also analyzed another baseline derived from the same GPT-2 model but trained on left-to-right (type from start of the sentence) words and trained on the same dataset. This comparison helps to highlight the advantage of using a bag-of-keywords.

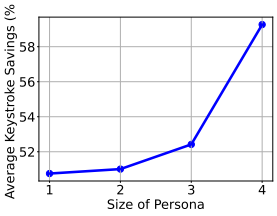
Figure 8 shows that as the number of keywords increases, the WER score decreases, corresponding to an improvement in performance. However, when more keywords are used, just entering the keywords alone yields a lower WER than the retrieved sentence. The reason for this stems from the output of a sentence retrieval system when trying to retrieve an unobserved sentence. For example, a golden reply can be "i've a bird and she loves cheese burgers like me, my favorite" and the retrieved reply may be "it does , but i love kids . what do you do ?". Table 5 shows that only the sentence generation model using bag-of-keywords produces practically meaningful responses, whereas TF-IDF produces responses that are only statistically meaningful. We observe that although the left-to-right generation model produces a sentence that is to some extent relevant to the question, the history is given greater importance in predicting the response than the starting words. It may be that these starting words are mostly stop words and lead to perturbations when training the model. Therefore, as there are more words input to the model, and although keywords may be present, the stop words out-weigh the keywords and confuse the language model, leading to a deteriorating result as shown in Figure 9. We can see in Figure 9 that the WER approaches 100%. These stop words have such a pronounced impact that the model is generating character sequences without space keys as shown in Table 5.

10 HUMAN JUDGMENT ANALYSIS

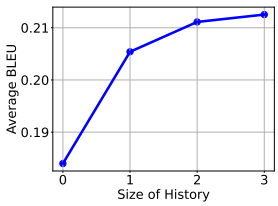
We complement our quantitative parameter analysis with an evaluation based on human judgment to assess the perceived quality of responses generated by KWickChat in terms of the semantic



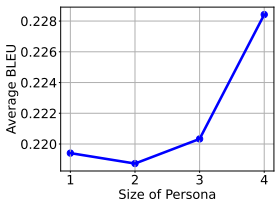
(a) Keystroke savings versus the size of history.



(b) Keystroke savings versus the number of persona tags.



(c) BLEU score versus the size of history.



(d) BLEU score versus the number of persona tags.

Figure 7: Benefits of increasing the size of history (number of exchanges in conversation history) and persona tags. More history and persona tags help the language generation model to output higher quality responses.

question: those are fun. i've a cat, do you?				
Methods	**User Input**	**Predicted Reply**	**Golden Reply**	WER (%)
Bag-of-Keywords	bird loves cheeseburgers like	i have a bird that loves cheeseburgers, i also like them	i've a bird and she loves cheeseburgers like me, my favorite	75
TF-IDF	bird loves cheeseburgers like	it does , but i love kids . what do you do ?	i've a bird and she loves cheeseburgers like me, my favorite	100
Left-to-Right	i've a bird and	abirdandsn't a cat, i've a dog	i've a bird and she loves cheeseburgers like me, my favorite	92

Table 5: A comparison of TF-IDF with sentence generation models using a bag-of-keywords and a left-to-right model.

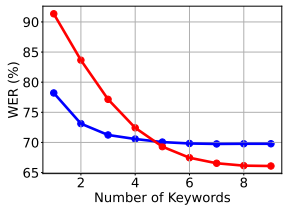


Figure 8: The blue line represents the average WER score against the number of keywords for TF-IDF. The red line represents the WER score based on raw keywords. The improvement from the increase in the number of keywords is marginal and soon saturates at four keywords.

Figure 9: Average WER score against the number of words for the left-to-right sentence generation model. This result suggest that this model is not usable as the generated sentences are not valid.

similarity between the responses and the golden reply. To provide source conversations for judgment, we used KWickChat to generate potential replies based on a set of conversations from the ChatAI2 dataset not used in training. Two volunteers were recruited as judges and instructed to rate the quality of the generated responses on a scale from 1 (very bad) to 5 (very good) using the golden reply as a reference. The ratings of the judges provide an alternative perspective that complements the WER and BLEU analysis, since we know these two metrics do not accurately reflect semantic similarity. The conversation text shown to the judges also included persona details. The two judges performed the task separately. 31 conversations were generated in total. The first conversation was used as a reference to explain the procedure. The next five conversations were designated for practice. The remaining 25 conversations served as the core for evaluation. Each conversation contained four exchanges, where each exchange was composed of one utterance from the speaking partner, one golden reply as the reference, and four generated responses as the suggested candidates. When generating the full set of conversations, and to accurately reflect the way in which KWickChat incorporates history, one of the generated responses was chosen at each exchange based on a consensus decision among three of the co-authors. Keywords were adopted from the BERT keywords extraction model to maximize reproducibility, although we do recognize that the keywords from BERT may not be fully representative of those chosen by actual AAC users.

We use quadratic weighted kappa as a metric for inter-rater reliability to measure the agreement between the two judges' ratings. This agreement typically varies from 0 (random agreement between

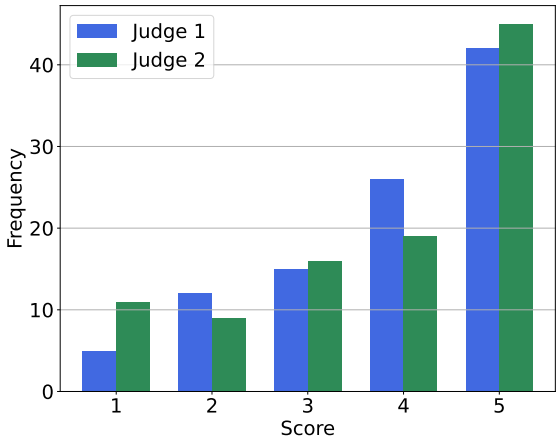


Figure 10: Bar plot of the highest score in each conversation exchange for both judges.

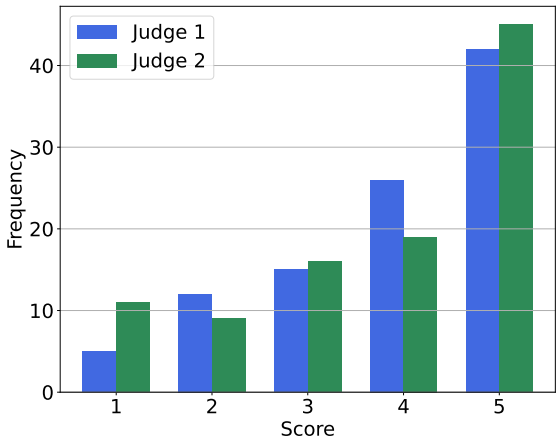


Figure 11: Bar plot of all the scores for the four generated responses in each conversation exchange for both judges.

raters) to 1 (complete agreement between raters). The two judges achieved 0.92 on the quadratic weighted kappa value, suggesting a strong agreement between the two judges. Figure 10 shows a bar plot of the highest score obtained among the four suggested replies for each exchange in the 25 conversations. The median score was 4 for both judges suggesting that most of the generated responses are of reasonable quality in terms of conveying the semantic information of the golden reply. Figure 11 shows the bar plot for all the scores given by judges on all the suggested replies. We observe in the two plots that although the frequency of a score of 1 is high in Figure 11, there are relatively few scores of 1 in Figure 10. This highlights the importance of multiple sentence suggestions and the variety of the results produced by the KWickChat language generation model.

A brief qualitative analysis was performed to assess in what circumstances generated responses were scored poorly by the judges. Most of the time this was because the golden reply was composed of several short sentences and the bag-of-keywords does not offer sentence break information. For example, one golden reply, "cute! where do you hike? my pet won't travel. she is a cow" has many sentence breaks with fairly loose semantic connections. It is currently not possible to provide punctuation information to indicate such sentence and semantic breaks. Including the ability to express punctuation is an important future feature to be explored for the KWickChat system.

11 EFFECT OF WORD PREDICTION AND SENTENCE DISPLAY

We now perform envelope analysis on the whole KWickChat system incorporating word prediction and sentence display functionality. This section serves to demonstrate the substantial keystroke savings that can be achieved by the system.

11.1 Word Prediction

Word prediction can help AAC users to substantially reduce keystrokes. Therefore, we investigate the impact of auto-complete on keystroke savings by examining word prediction accuracy as a controllable parameter (when enabled, default set to 0.6). This is the simulated probability that auto-complete is successful. Here word prediction accuracy therefore determines the probability that for a given keystroke the intended word would be auto-completed. Kristensson et al. [15] used this model as a simplification of a true auto-complete assistance predicting the intended word as being the same regardless of how many keystrokes the user has typed. Figure 12a shows that keystroke savings are substantially improved using word prediction when word prediction accuracy is set to 0.6. Figure 12b illustrates that keystroke savings increase as prediction accuracy increases. We see that with word prediction and with a threshold word error rate of 0.65, the keystroke savings of the KWickChat system is around 71%.

11.2 Sentence Display

Multiple sentences can be generated from the KWickChat language generation model by changing the random seed in the sampling algorithm. Table 1 illustrates the importance of displaying a variety of generated sentences. The difference between the generated sentences decreases when there is a larger size of bag-of-keywords as the increase in the number of keywords strengthens the model's belief in the output sentence. We identify the controllable parameter here to be the number of sentences generated/displayed. Figure 12c shows the improvement brought by the number of sentences displayed. We see that improvement saturates when the number of sentences displayed approaches five. There is a trade-off in keystroke savings and the number of displayed sentences as displaying more sentences adds cognitive overhead on users, which may eventually negatively affect entry rates. An in-depth analysis of this trade-off (in a word prediction context) is available in prior work [16].

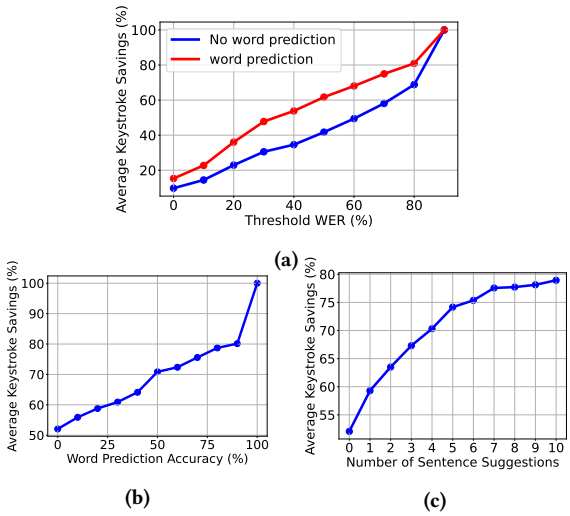


Figure 12: Envelope analysis on word prediction and sentence display. Both word prediction and sentence display play an important role in improving the performance of the KWickChat system. a) WER scores change as the threshold WER changes for word prediction and no word prediction. b) Keystroke savings versus word prediction accuracy. c) Keystroke savings versus the number of sentences suggested.

12 LIMITATIONS AND FUTURE WORK

Exciting future work is to investigate how well KWickChat works with AAC users, their friends and families and personal assistants. Successful use of new technology relies on a deep understanding of this entire AAC "ecosystem". We anticipate work in eliciting requirements relating to the specifics of user interface design, such as *how* sentences are presented, and discussions with end-users about agency and locus of control in conversations, given that there is a risk this system "puts words into people's mouths". Such rich studies with end-users are possible now that a working system has been completed and demonstrated in terms of efficacy in envelope analysis. We hope this work, and the fact that we share the trained model, will stimulate further work in the area of high-quality sentence-generation for AAC dialogues.

One failure mode of KWickChat identified in this study is that the model may not correctly predict a golden reply even when all the keywords in the sentence have been entered. In practice, this outcome is unlikely if a user inputs relevant keywords. However, when such a failure does occur it may add significant cognitive load on the user, and incur an additional time cost, if the user then tries to manually enter the remaining stop words to complete the sentence. As future work, we plan to address this failure mode by incorporating a simple sentence completion model that can insert stop words between keywords to construct a full sentence. This sentence with injected stop words can then be presented in parallel with the other generated replies.

More generally there are a number of avenues to explore for improving the overall performance of KWickChat. One interesting potential strategy is to leverage a Bayesian neural network that would allow the system to adapt to the user over time. A similar outcome might also be achieved by streamlining the process of retraining the language generation model with newly observed conversational data. There are also likely potential benefits in supporting automated extraction of persona information for both the user and the speaking partners. By incorporating more persona details, the system is likely to produce more user-specific and speaking partner-specific responses. Finally, we envisage extending KWickChat to support multi-party AAC dialogue generation, which would allow the user to dynamically produce cohesive responses efficiently when there are multiple speaking partners.

13 CONCLUSIONS

In this paper we have demonstrated that KWickChat can generate meaningful replies in daily conversations using very little user input. The extensive quantitative envelope analyses presented serve to highlight how our novel use of a bag-of-keywords, conversation history and persona details can enhance the quality of generated responses. These features combined with word prediction and the display of multiple alternatives result in potential keystroke savings of 77% assuming a WER acceptance threshold of 0.65. The evaluation by two human judges also indicates that the KWickChat system is able to produce sentences that are semantically consistent with target reference sentences. The median for both judges of the highest scoring reply in 100 exchanges across 25 conversations was 4 on a scale from 1 (very bad) to 5 (very good). The capabilities and performance of KWickChat therefore suggest that it has the potential to reduce the communication gap experienced by AAC users. Now that the potential of KWickChat has been established, it is viable to move forward with the exciting next steps of engaging AAC users in co-design and evaluation and we encourage designers and researchers to join us in building on this work using our code and trained models.

OPEN SCIENCE

Complete source code for KWickChat and the trained language generation model can be found here: https://github.com/CambridgeIIS/KWickChat/.

REFERENCES

[1] Carlo Aliprandi, Nicola Carmignani, Nedjma Deha, Paolo Mancarella, and Michele Rubino. 2008. Advances in nlp applied to word prediction. *University of Pisa, Italy February* (2008).
[2] BR Baker. 2002. Rate and augmentative communication devices: Symantec compaction, RESNA (Rehabilitation Engineers Society of North America) 25th International Conference.
[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).
[4] Boxing Chen and Colin Cherry. 2014. A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Baltimore, Maryland, USA, 362–367. https://doi.org/10.3115/v1/W14-3346
[5] Ann Copestake. 1997. Augmented and alternative NLP techniques for augmentative and alternative communication. In *Natural Language Processing for Communication Aids*.
[6] Patrick W. Demasco and Kathleen F. McCoy. 1992. Generating Text from Compressed Input: An Intelligent Interface for People with Severe Motor Impairments. *Commun. ACM* 35, 5 (May 1992), 68–78. https://doi.org/10.1145/129875.129881
[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[8] Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al.

2019. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098* (2019).
[9] Alexander Fiannaca, Ann Paradiso, Mira Shah, and Meredith Ringel Morris. 2017. AACrobat: Using mobile devices to lower communication barriers and provide autonomy with gaze-based AAC. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 683–695.
[10] Google. 2013. word2vec. https://code.google.com/archive/p/word2vec/.
[11] Maarten Grootendorst. 2020. KeyBERT: Minimal keyword extraction with BERT. https://doi.org/10.5281/zenodo.4461265
[12] D Jeffery Higginbotham, Ann M Bisantz, Michelle Sunm, Kim Adams, and Fen Yik. 2009. The effect of context priming and task type on augmentative communication performance. *Augmentative and Alternative Communication* 25, 1 (2009), 19–31.
[13] Jeffery Higginbotham, Gregory Lesher, Bryan Moulton, and Brian Roark. 2011. The Application of Natural Language Processing to Augmentative and Alternative Communication. *Assistive technology : the official journal of RESNA* 24 (04 2011), 14–24. https://doi.org/10.1080/10400435.2011.648714
[14] Arlene W Kraat. 1987. Communication interaction between aided and natural speakers: A state of the art report. (1987).
[15] Per Ola Kristensson, James Lilley, Rolf Black, and Annalu Waller. 2020. *A Design Engineering Approach for Quantitatively Exploring Context-Aware Sentence Retrieval for Nonspeaking Individuals with Motor Disabilities*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376525
[16] Per Ola Kristensson and Thomas Müllners. 2021. Design and Analysis of Intelligent Text Entry Systems with Function Structure Models and Envelope Analysis. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.
[17] Stefan Langer and Marianne Hickey. 1998. Using Semantic Lexicons for Full Text Message Retrieval in a Communication Aid. *Nat. Lang. Eng.* 4, 1 (March 1998), 41–55. https://doi.org/10.1017/S1351324998001855
[18] Janice Light. 1988. Interaction involving individuals using augmentative and alternative communication systems: State of the art and future directions. *Augmentative and alternative communication* 4, 2 (1988), 66–82.
[19] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. 13. https://doi.org/10.18653/v1/D16-1230
[20] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023* (2016).
[21] Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149* (2017).
[22] Conor McKillop. 2018. Designing a context aware AAC solution. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. 468–470.
[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[24] Oluwatobi Olabiyi, Alan Salimov, Anish Khazane, and Erik T Mueller. 2018. Multi-turn dialogue response generation in an adversarial learning framework. *arXiv preprint arXiv:1805.11752* (2018).
[25] OpenAI. 2013. OpenAI GPT-2: 1.5B Release. https://openai.com/blog/gpt-2-1-5b-release/.
[26] Jahna Otterbacher, Gunes Erkan, and Dragomir Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*. 915–922.
[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.
[28] Rupal Patel and Rajiv Radhakrishnan. 2007. Enhancing Access to Situational Vocabulary by Leveraging Geographic Context. *Assistive Technology Outcomes and Benefits* 4, 1 (2007), 99–114.
[29] Felix Putze, Tilman Ihrig, Tanja Schultz, and Wolfgang Stuerzlinger. 2020. Platform for Studying Self-Repairing Auto-Corrections in Mobile Text Entry based on Brain Activity, Gaze, and Context. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[30] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).
[31] Ehud Reiter, Ross Turner, Norman Alm, Rolf Black, Martin Dempster, and Annalu Waller. 2009. Using NLG to help language-impaired users tell stories and participate in social dialogues. *Proceedings of the 12th European Workshop on Natural Language Generation, ENLG 2009* (01 2009). https://doi.org/10.3115/1610195.1610196
[32] Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation* (2004).
[33] Igor Schadle. 2004. Sibyl: AAC system using NLP techniques. In *International Conference on Computers for Handicapped Persons*. Springer, 1009–1015.
[34] Edward Schofield and Gernot Kubin. 2002. On interfaces for mobile information retrieval. In *International Conference on Mobile Human-Computer Interaction*. Springer, 383–387.
[35] J Todman, N Alm, P File, and J Higginbotham. 2004. *An office workplace prototype for an utterance-based communication aid for people without speech: Extracts from Objective 1(ii) evaluation, Objective 2 and Objective 3*. Technical Report. Extract retrieved September 2007 from: http://www.dundee.ac.uk/psychology/jtodman.
[36] John Todman, Norman Alm, Jeff Higginbotham, and Portia File. 2008. Whole utterance approaches in AAC. *Augmentative and alternative communication* 24, 3 (2008), 235–254.
[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[38] Horabail Venkatagiri. 1999. Efficient keyboard layouts for sequential access in augmentative and alternative communication. *Augmentative and Alternative Communication* 15, 2 (1999), 126–134.
[39] Keith Vertanen and Per Ola Kristensson. 2011. The imagination of crowds: conversational AAC language modeling using crowdsourcing and large data sources. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 700–711.
[40] Ellen M Voorhees. 1999. Natural language processing and information retrieval. In *International summer school on information extraction*. Springer, 32–48.
[41] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745* (2015).
[42] Thomas Wolf. 2019. *How to build a State-of-the-Art Conversational AI with Transfer Learning*. https://medium.com/huggingface/how-to-build-a-state-of-the-art-conversational-ai-with-transfer-learning-2d818ac26313
[43] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. *CoRR* abs/1901.08149 (2019). arXiv:1901.08149 http://arxiv.org/abs/1901.08149
[44] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243* (2018).
[45] Zhuosheng Zhang and Hai Zhao. 2021. Advances in Multi-turn Dialogue Comprehension: A Survey. *arXiv preprint arXiv:2103.03125* (2021).