

University of Dayton eCommons

Computer Science Faculty Publications

Department of Computer Science

1-5-2018

Natural Language, Mixed-Initiative Personal Assistant Agents

Joshua W. Buck

University of Dayton, jbuck1@udayton.edu

Saverio Perugini

University of Dayton, sperugini1@udayton.edu

Tam W. Nguyen

University of Dayton, tnguyen1@udayton.edu

Follow this and additional works at: https://ecommons.udayton.edu/cps_fac_pub



Part of the [Graphics and Human Computer Interfaces Commons](#), and the [Other Computer Sciences Commons](#)

eCommons Citation

Buck, Joshua W.; Perugini, Saverio; and Nguyen, Tam W., "Natural Language, Mixed-Initiative Personal Assistant Agents" (2018). *Computer Science Faculty Publications*. 174.

https://ecommons.udayton.edu/cps_fac_pub/174

This Conference Paper is brought to you for free and open access by the Department of Computer Science at eCommons. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of eCommons. For more information, please contact frice1@udayton.edu, mschlangen1@udayton.edu.

Natural Language, Mixed-initiative Personal Assistant Agents

Joshua W. Buck, Saverio Perugini, Tam V. Nguyen

Department of Computer Science

University of Dayton

Dayton, Ohio 45469-2160 USA

{jbuck1,saverio,tamnguyen}@udayton.edu

ABSTRACT

The increasing popularity and use of personal voice assistant technologies, such as *Siri* and *Google Now*, is driving and expanding progress toward the long-term and lofty goal of using artificial intelligence to build human-computer dialog systems capable of understanding natural language. While dialog-based systems such as *Siri* support utterances communicated through natural language, they are limited in the flexibility they afford to the user in interacting with the system and, thus, support primarily action-requesting and information-seeking tasks. Mixed-initiative interaction, on the other hand, is a flexible interaction technique where the user and the system act as equal participants in an activity, and is often exhibited in human-human conversations. In this paper, we study user support for mixed-initiative interaction with dialog-based systems through natural language using a *bag-of-words* model and *k-nearest-neighbor* classifier. We study this problem in the context of a toolkit we developed for automated, mixed-initiative dialog system construction, involving a dialog authoring notation and management engine based on lambda calculus, for specifying and implementing task-based, mixed-initiative dialogs. We use ordering at *Subway* through natural language, human-computer dialogs as a case study. Our results demonstrate that the dialogs authored with our toolkit support the end user's completion of a natural language, human-computer dialog in a mixed-initiative fashion. The use of natural language in the resulting mixed-initiative dialogs afford the user the ability to experience multiple self-directed paths through the dialog and makes the flexibility in communicating user utterances commensurate with that in dialog completion paths—an aspect missing from commercial assistants like *Siri*.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; *Natural language processing*; • **Human-centered computing** → *Human computer interaction (HCI)*; *Natural language interfaces*;

This document is the authors' accepted manuscript, provided in compliance with ACM's policy on self-archiving.

Presented at *IMCOM '18, January 5–7, 2018, Langkawi, Malaysia, and published in the conference proceedings.*

© 2018 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

KEYWORDS

Bag of words model; dialog management; function currying; human-computer dialogs; interactive voice response systems; k-nearest-neighbor classifier; lambda calculus; mixed-initiative dialogs; mixed-initiative interaction; natural language processing; partial evaluation.

ACM Reference Format:

Joshua W. Buck, Saverio Perugini, Tam V. Nguyen. 2018. Natural Language, Mixed-initiative Personal Assistant Agents. In *IMCOM '18: The 12th International Conference on Ubiquitous Information Management and Communication, January 5–7, 2018, Langkawi, Malaysia*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3164541.3164609>

1 INTRODUCTION

Human-computer dialogs, which are used to improve information access in smart phone apps, ATMs, and airport kiosks are woven into the fabric of our daily interactions with computer systems. The increasing popularity and use of personal voice assistant technologies, such as *Siri*, *Google Now*, *Cortana*, and *Alexa*, is driving and expanding progress toward the lofty and long-term goal of using artificial intelligence to build human-computer dialog systems capable of understanding natural language [22]. Dialog-based systems can be classified based on the degree of *flexibility* and *natural language* supported (see Figure 2).

1.1 Flexibility in Dialog: Mixed-initiative Interaction

A form of flexibility in human-computer dialog is mixed-initiative interaction. *Mixed-initiative interaction* is a flexible interaction strategy whereby the user and the system engage as equal participants in an activity and take turns exchanging initiative as the user progresses toward the satisfaction of a particular goal facilitated by her interaction with the system [16]. Mixed-initiative interaction is often exhibited in human-human conversations, and has been identified and studied as a link to bridge artificial intelligence and human-computer interaction [16]. The sample mixed-initiative, human-computer dialog in Figure 1 between a user, interested in ordering a sandwich at the restaurant *Subway*, and the dialog agent illustrates both the flexibility in dialog afforded by mixed-initiative interaction as well as the complexities in its implementation due to the numerous and varied directions in which the user might steer the dialog.

- In line (1), the agent prompts for choice of sub or salad. In line (2), the user responds directly to the prompt. This *system-initiated, fixed* mode of interaction is common most dialog systems today.

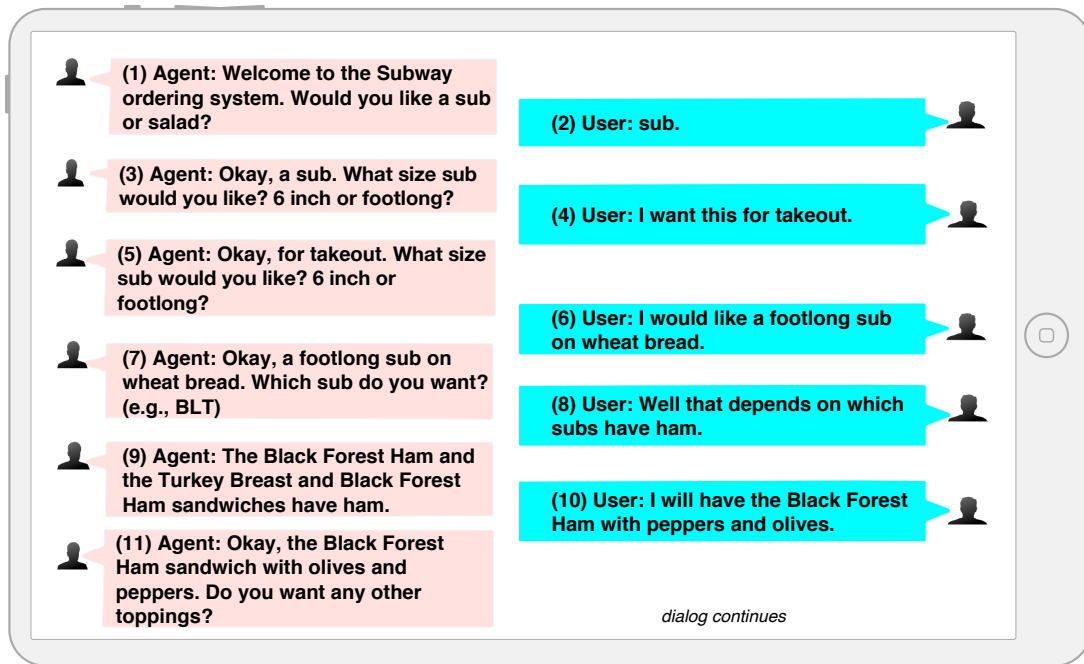


Figure 1: Sample motivating, mixed-initiative dialog, built with our toolkit, running on a mobile device.

- In line (3), the agent solicits for the next item in the script—sandwich size. In line (4), the user, however, does not respond directly to the prompt. Rather, the user provides unsolicited information (i.e., the order is for take out) that is relevant to the discussion at hand. Such out-of-turn responses enable the user to take the dialog initiative and support a tier of mixed-initiative interaction called *unsolicited reporting* [1].
- In line (5), the agent, again, prompts for sandwich size—information which has yet to be provided by the user. In line (6), the user responds to the prompt for size, but also specifies the type of bread desired—more unsolicited, but relevant, information—in one stroke. Providing more than one response in a single dialog turn is another aspect of mixed-initiative interaction.
- In line (7), the agent accepts the user information from line (6), and inquires which type of sub the user desires. In line (8), rather than providing information to the agent, the user seeks information from the agent—which subs have ham.
- In line (9), the agent processes the user’s request for information and provides the names of specialty sandwiches with ham. The user chooses one of the items with ham and also specifies a desire for the toppings of peppers and olives.
- In line (11), the agent accepts the user information and inquires if the user wants any additional toppings.

The Subway dialog in Figure 1 illustrates a mixed-initiative, human-computer dialog that, due to the complexity and variability in the dialog, is not possible to realize *through natural language* with other dialog systems today. Since “authoring a dialogue is like writing a movie script with many different endings” [19], “a central problem for mixed-initiative dialogue management is coping with utterances

that fall outside of the expected sequence of the dialogue” [29]. Trying to support all permutations and combinations of responses to just three questions results in 8,191 possible unique paths through the dialog [26].

We built a dialog system construction toolkit (see Figure 6, left), involving a dialog authoring notation and dialog management engine, which supports all of these forms of mixing initiative (in Figure 1), through the specification of keywords (e.g., ‘large’), rather than natural language [26]. The system now supports the use of natural language using an autonomously generated dictionary and a bag-of-words model.

1.2 Natural Language in Dialog

Dialog-based systems such as *Siri* support utterances communicated through natural language, but are limited in flexibility to action-requesting, information-seeking (e.g., “What is the weather forecast tomorrow?”), and information-providing utterances and, thus, only support a low degree of flexibility (see Figure 3). In this paper, we study user support for mixed-initiative interaction with dialog-based systems through natural language using a *bag-of-words* model and *k-nearest-neighbor* classifier. *Our research goal to support users’ interaction with dialog-based systems like Siri in a mixed-initiative mode through the use of natural language* (see Figures 4 and 5, and lower right corner of Figure 2). Alternatively said, our research goal is to support the use of (a high degree of) natural language utterances in the mixed-initiative dialog systems created using our dialog construction toolkit. The use of natural language in the resulting mixed-initiative dialogs affords a user the ability to experience multiple self-directed paths through the dialog and makes the flexibility in communicating user utterances

		<i>Degree of Natural Language</i>	
		no NL ←	→ complete NL
<i>Degree of Flexibility (MII)</i>	fixed	voice commands	<i>Stanford CoreNLP</i> [23]
	↑		
	⋮	<i>Apple's Siri</i>	
	↓		<i>NADIA</i> [3]
	mixed	our prior research [6, 25, 26] ~> our current research [7]	

Figure 2: A design space for dialog-based systems [7].

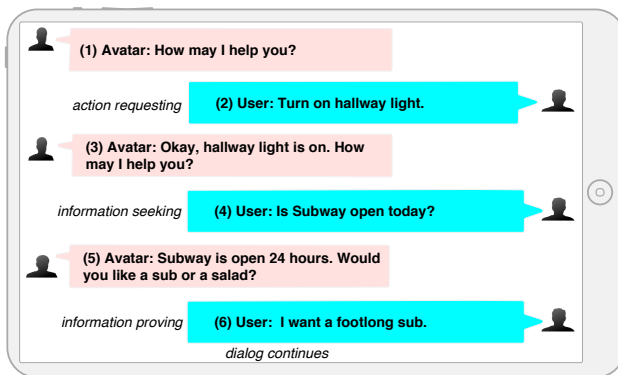


Figure 3: User interactions with dialog systems like *Siri* are rigid and limited (in flexibility through natural language) to action-requesting, information-seeking, and information-providing utterances.

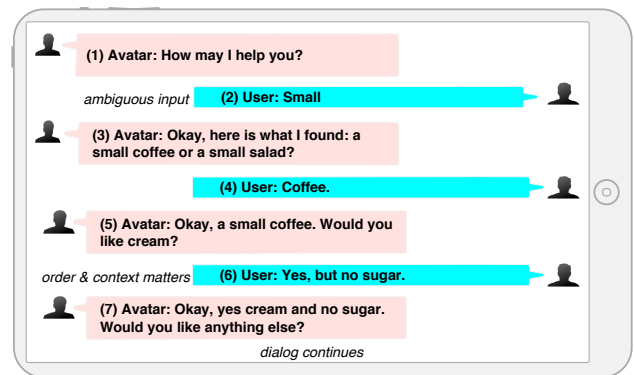


Figure 5: Demonstration of the flexibility in natural language that our approach fosters beyond that provided by personal assistants like *Siri*.

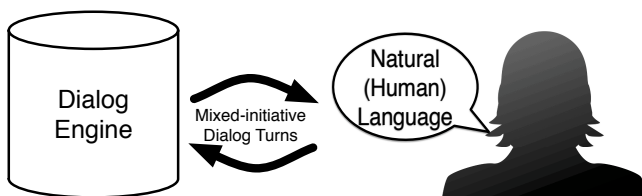


Figure 4: Graphical overview of participant relationship in a human-computer dialog.

commensurate with that in dialog completion paths—an aspect missing from commercial assistants like *Siri*. In short,

$$\text{Natural Language} + \text{Mixed-initiative Interaction} = \text{Natural \& Flexible Human-Computer Dialog}$$

Our enhanced dialog toolkit, reported in this paper (see Figure 6), supports all of the forms of mixing initiative, demonstrated in Figure 1, through the use of natural language.

2 BACKGROUND AND RELATED RESEARCH

There are two main approaches to dialog modeling and management: task-based and data-driven. Our work combines the two: (i)

our research targets task-based dialog systems whose goal is to support the user in satisfying clearly-defined goals by completing highly-structured tasks (see §§ 2.1–2.2); and (ii) we use data-driven techniques (e.g., *bag-of-words* model and a *k-nearest-neighbor* classifier) to support the users’ use of natural language to pursue these tasks (see § 3 and Figure 7).

2.1 Dialog Authoring Notation

“The task-based approach involves modeling a collection of tasks to be supported by the dialog system, using a modeling notation, and discerning how the user can be most effectively afforded (the desired) interaction flexibility in completing those tasks. Finite state automata, and other transition networks, context-free grammars, and events have been used as general task structures to model dialog [14]” [26]. Other task modeling approaches involve agenda [28] and rule-oriented structures [11].

Our approach, on the other hand, is unique in that we use concepts and operators from programming language theory, and in particular lambda-calculus, rather than task structures, to model dialog. “We designed a notation based on lambda calculus [12] ... as an authoring notation for specifying dialogs and also suggests implementation ideas [(discussed below). This] distinguishes our model from other knowledge/task-based approaches which use

hierarchical task/agenda models” [26]. In particular, our approach to dialog modeling involves thinking of dialog as a function and uses concepts from lambda-calculus, including function currying and partially evaluation [18], to automatically modify that function to achieve a mixed-initiative mode of interaction. “As the user progresses through a dialog, we think of the steps that she takes as the evaluation of a function. Changing the evaluation method of the function (or transforming the function) then corresponds to different interaction policies [27] for the dialog (i.e., ways of mixing initiative).” The “overall idea is that different function evaluation strategies correspond to different interaction policies for the dialog (i.e., system initiated vs. mixed initiative) or ways of mixing initiative” [6]. For the details of the notation, we refer the reader to [26].

Our use of concepts and operators from theoretical programming languages, rather than task structures, to model dialog, has a host of advantages. For instance, our language-based notation for specifying dialogs is rich enough to capture multiple orders of responses (e.g., lines 3 and 4 in Figure 1) independent of multiple responses per utterance (e.g., line 6 in Figure 1) as well as sub-dialogs. Moreover, our notation lends itself to evaluation through the number of sub-expressions in the notation necessary to capture an entire dialog specification—an evaluation metric which also measures the control complexity of the implementation in an implementation-neutral way. Furthermore, “the structure of an expression in our dialog-authoring notation and the language concepts used therein, capturing the requirements of a dialog, provide a pattern for implementing the dialog. [Thus, the concepts from lambda-calculus] are not just helpful metaphors for dialog specification, but also lend insight into operationalizing dialogs. [We use concepts and operators from lambda calculus to] intensionally model multiple paths through a dialog without extensionally hardcoding each into the control flow of the implementation” [26]. Based on this theoretical foundation, we built a dialog management engine which is capable of automatically realizing a variety of mixed-initiative dialogs given only a single, high-level specification in our notation of each.

2.2 Dialog Management Engine

The *dialog management* component of a dialog-based system is concerned with determining what to prompt for and/or accept next based on what has already been communicated and the current utterance [20]. “The dialog management component plays a central role in the architecture of a traditional dialog system, and is primarily concerned with controlling the flow of the dialog while maintaining discourse history [20], sometimes referred to as system-action prediction, and coordinating with other (typically input/output) components of the system (e.g., automatic speech recognition, spoken language understanding, and presentation of results)” [26]. “One of the most time consuming aspect of dialogue system development today is the implementation of the dialogue manager” [13].

Our toolkit factors the domain-dependent components of a dialog system (e.g., the aspects of the dialog specific to the targeted domain) from the domain-independent components of the system (e.g., the dialog engine and management). Figure 6 (left) depicts the architectural design of our Dialog Management Engine and

illustrates the independence of the Dialog Staging Engine from the generated XML specification of the dialog that it operationalizes. Thus, our dialog staging engine “acts as an interpreter, in the programming languages sense, for the given dialog specification. [This approach provides] a clean separation of the domain-dependent and -independent aspects (e.g., control logic and dialog flow) [2]. This approach is used in the *RavenClaw* dialog management framework [4, 5]. *RavenClaw* uses an agenda-based approach to task modeling [27, 28]. Our framework is an instantiation of this ‘separation of task model and dialog engine’ approach to dialog management (see Figure 6, left)” [26]. Our dialog management engine can automatically operationalize a dialog that involves multiple prompts and/or sub-dialogs, given a high-level dialog specification of it in our notation.

2.3 Putting it All Together

Combining our *dialog authoring notation* with the *dialog management engine*, whose independence from a dialog specification authored in that notation is fostered by the use of lambda-calculus in the notation, yields a toolkit for automating the construction of mixed-initiative dialog systems. Thus, we generalized and automated the activity of building a dialog system. While “developing a mixed-initiative dialog system is a complex task” [17], “creating an actual dialog system involves a very intensive programming effort” [15], and “complete automation in creating ... dialog applications remains an extremely difficult problem” [10], given a specification of a mixed-initiative dialog in our dialog authoring notation, our dialog engine automates the implementation/realization of it. With our toolkit, we can generate dialog systems for a variety of unsolicited reporting, mixed-initiative dialogs. In short, we have built a toolkit for specifying and implementing task-based, mixed-initiative, human-computer dialogs. Our dialog system construction toolkit is available for download at https://bitbucket.org/jwb_research.

Other research projects seeking to automate the implementation of flexible, dialog-based systems include [17, 19]. However, using concepts and operators from lambda calculus to specify and operationalize dialogs “is a fundamentally different approach to dialog modeling, management, and implementation” [26]. With this foundation in place we now address the incorporation of natural language into our model.

3 TECHNICAL DETAILS: OUR APPROACH

3.1 Bag-of-Words Model

Figure 5 demonstrates the natural language capabilities of our mixed-initiative dialog system. We have enhanced our model for mixed-initiative dialog by using a *bag-of-words* model for a new dialog domain. In particular, we parse each sentence in the dialog into the corresponding feature vector representing the frequencies of each meaningful word. (Common stop words including ‘a’, ‘an’, and ‘the’ are discarded.) Each bag-of-word feature vector is then normalized to ℓ_2 norm. Next, we adopt a *k-nearest-neighbor* classifier with Euclidean distance to predict the context of a user utterance (i.e., map an unsolicited utterance to the dialog prompt to which it is a response) to improve the natural language and mixed-initiative capabilities of systems like *Siri*.

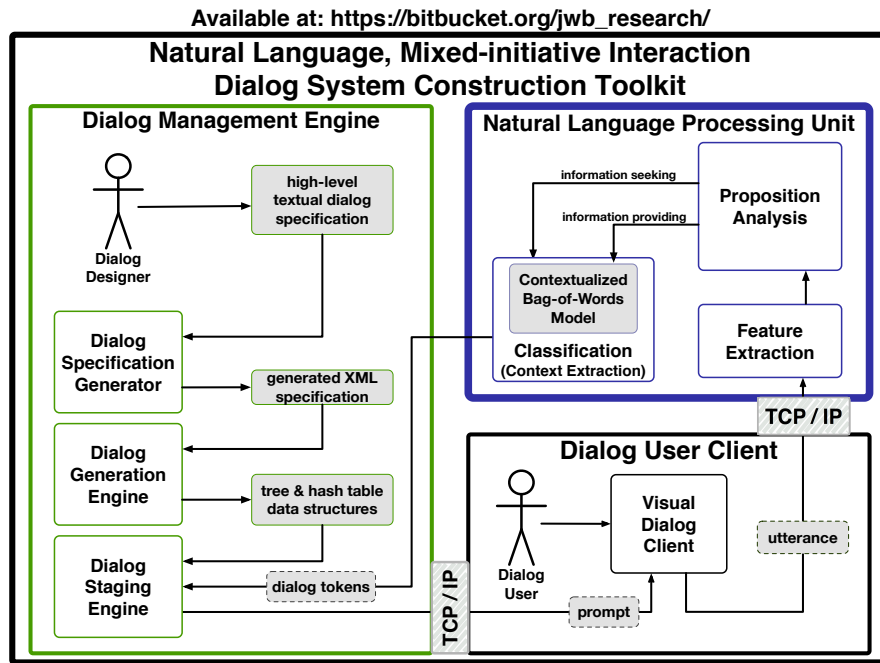


Figure 6: Architecture of our natural language, mixed-initiative dialog construction toolkit, highlighting the Natural Language Processing Unit—the main contribution of this paper [7].

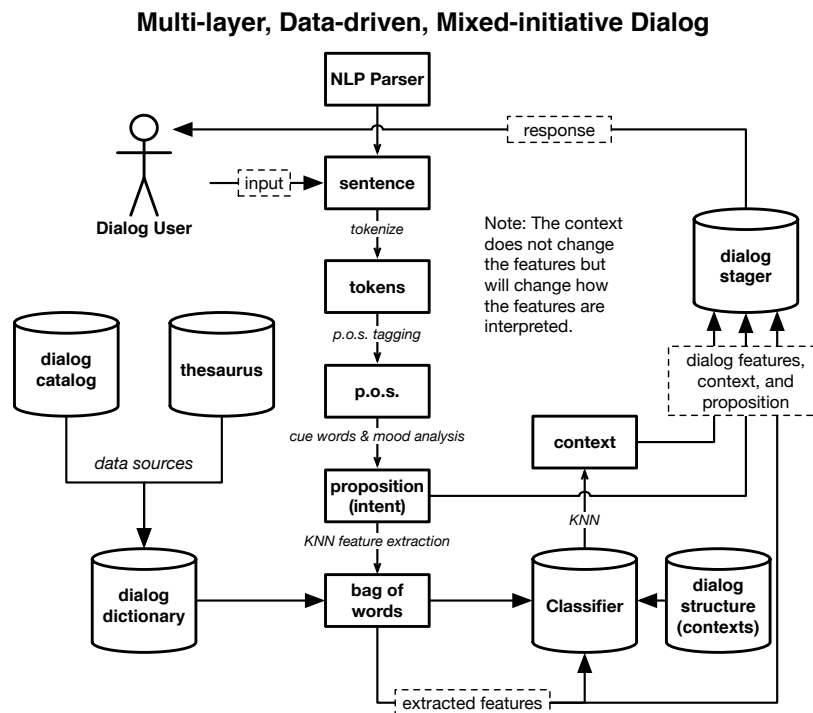


Figure 7: Detailed data-driven architectural design of our Natural Language Processing Unit in Figure 6 [7].



Figure 8: Data structure representation of our bag-of-words model.

Figure 7 illustrates the data-driven design of the Natural Language Processing Unit in Figure 6. The center column of Figure 7, from the top-most box labeled ‘NLP Parser’ down to the bottom-most box labeled ‘bag of words,’ depicts a typical NLP strategy for analyzing user utterances. We use the open-source Natural Language Toolkit library (NLTK; see <http://www.nltk.org/>) [3] in this part of our system. Our bag-of-words model is generated from the combination of a dialog catalog, a word and phrase thesaurus, and other data sources such as a database, if necessary. The dialog catalog is a list of all of the words in and extracted from a given domain (e.g., ordering from Subway). Once complete, the bag-of-words model becomes part of a more complex classifier with three objectives; (i) to identify dialog tokens, (ii) to identify the context in which the dialog tokens fit, and (iii) to determine if the user is providing information, seeking information, or extracting information. The bag-of-words model serves to fulfill the first objective of identifying probable dialog tokens. The prior user responses coupled with the dialog structure, automatically generated from a given dialog expression, provide the basis for determining the current context(s) of these tokens. We discern the intent or proposition of the user using the proposition analysis capabilities in NLTK. These natural language processing techniques provide the input to the dialog stager, which validates the extracted dialog tokens in the identified contexts with the given user’s intention and provides an appropriate response.

In summary, our Natural Language Processing Unit includes

- use of NLTK for tokenization and part-of-speech tagging;
- proposition analysis as outlined in [3];
- a bag-of-words model for parsing the most probable dialog tokens from an utterance;
- rules to break ties from the *k-nearest-neighbor* classifier;
- rule-based context analysis using past and present utterances; and
- automatic addition of similar words and phrases to those extracted from the dialog catalog using a thesaurus.

Our toolkit for automatic, natural language, mixed-initiative dialog system construction is available at https://bitbucket.org/jwb_research.

3.2 Case Study: Subway

As a case study of a dialog system designed with our toolkit, we modeled the menu at the popular sandwich shop *Subway*.

The menu we modeled is available at http://local.subway.com/FWPCons/MenuPDF/USA_Menu.pdf. We modeled a subset of this Subway menu with 32 dialog words (e.g., ‘wheat,’ ‘footlong,’ and ‘American cheese’). There are 167 synonyms for these 32 dialog words that are automatically added to the domain using a word and phrase thesaurus. In this case study, 4,801 unique ways to order at Subway are possible. However, through the use of mixed-initiative interaction, there are 4,420,080 ways to arrive at each of those possible 4,801 Subway orders and, thus, 21,220,804,080 possible dialog completions in total. In our dataset, there are effectively 32 vectors in the bag-of-words model—one vector for each dialog key. Thus, each vector contains 199 entries as there are 32 + 167 words in the dialog. However, no table is stored as the ‘hashing trick’ is used in place of a physical table or vector structure. The Bag-of-Words (BoW) model simply picks the dialog key for which the most words are submitted. This leads to many ties. For instance, if the user only says ‘Italian,’ the BoW model equally scores ‘Italian Sandwich’ and ‘Italian Dressing’ as likely assuming that no other information is given (see Figure 8 which corresponds to this ‘Italian’ example). If it is known that the user is talking about salads, then ‘Italian dressing’ is more likely. In these instances, the context is considered when scoring. Thus, if ‘Italian Sandwich’ is a response to the current solicitation and the user only says the utterance ‘Italian,’ it is assumed that the user utterance was ‘Italian Sandwich.’ Furthermore, there are currently over 94 tie breaking rules in place which consider the current context, what information has already been provided, and what is most commonly answered. We also use a variety of strategies to train the system to make improved guesses and correctly break more ties, based on users’ repeated evaluation of a particular dialog domain. In particular, rather than prompting the user for clarification (e.g., “Did you mean Italian dressing or an Italian sub?”), we incorporate controls to direct the system to what degree it can make inferences.

4 EVALUATION

Evaluating models for mixed-initiative dialog is itself an unsolved problem for a variety of reasons including the extremely limited nature of existing data and the ambiguity of the definition of initiative. One way to capture the efficacy of a model is to evaluate how well the model fits data. In the context of our model, this means evaluating the frequency of dialogs that can be captured by our programming language-based, dialog-authoring notation and how well it captures each. Specifically, given q , the number of questions

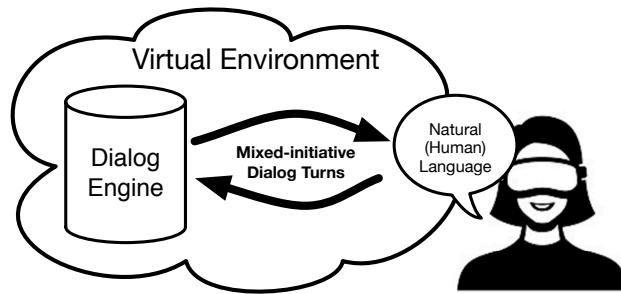


Figure 9: Our future research: use of natural language, mixed-initiative dialog within virtual environments.

posed in a dialog, our system is capable of automatically implementing $\sum_{p=1}^q p! \times S(p, q)$ dialog specifications (e.g., 8,192 for $q = 3$). In our study, we found that over 20% of the dialogs (1,692/8,192) can be compressed 50% or more [26]. We evaluated the descriptive and staging capabilities of our toolkit by demonstrating that it can succinctly capture and operationalize a wide variety of dialogs, including those involving sub-dialogs.

We also have demonstrated both our dialog toolkit for automatic human-computer dialog system construction as well as the resulting dialog systems to numerous computer science faculty, researchers, staff, and students at a variety of national and international conferences [6, 7, 25, 26] and informal feedback has been overwhelmingly positive. In particular, in March 2017 the presentation and demonstration of our dialog toolkit resulted in a first-place victory in the *2017 ACM student research competition* [7] (see <https://src.acm.org/winners/2017>). These results demonstrate that the dialogs authored with our toolkit support the end user’s completion of a human-computer dialog in a manner that is natural and resembles human-human interaction (i.e., in a mixed-initiative fashion). We are currently designing usability studies with users to gather formal results.

5 CONCLUSION

Dialog has been established as an effective mechanism through which to achieve a rich form of human-computer interaction [8]. Dialog-based systems are now used in domains as critical as health care [24]. The ability to automatically create a dialog system in a new domain is important. We feel that i) a *mixed-initiative mode of interaction* driven by user utterances and ii) communicated through the use of *natural language* (see lower right hand cell of Figure 2) is the key to the effectiveness and widespread adoption of personal assistant technologies. This paper discusses a research project that marries (i) and (ii).

5.1 Contributions

We have extended our previous work [6, 26] by introducing the Natural Language Processing Unit (see Figure 7) into our mixed-initiative dialog toolkit (see Figure 6). This involved addressing the challenge of reconciling the use of natural language with mixed-initiative interaction. The addition of the NLP unit fosters flexibility in the use of natural language commensurate with the flexibility

in dialog completion paths in the mixed-initiative dialog systems constructed with our toolkit. The challenges involved included addressing the increased ambiguity of the utterances provided by the user due to the explosion in the number of dialog completion paths enabled by mixed-initiative interactions as well as the necessity to establish a context or set of contexts in which the user and the system operate. The system also can now discern the user’s intent to distinguish between information-providing and information-seeking utterances. We also enhanced the degree of mixed-initiative interaction supported by empowering the user to make meta-dialog inquiries of the system within some context (as seen in line 8 of Figure 1). The dialog system is now able to extract information from a dialog domain and process the current context as well as all of the information the user has already provided. This approach reduces redundancies and provides the user with an understanding of what her options are vis-à-vis the current state of the dialog.

We are optimistic that our toolkit can have an impact the development of interactive systems where flexibility in human-computer dialog is important, especially using natural language. For instance, designers of task-based, natural language, mixed-initiative dialog systems can use our dialog authoring notation and engine as a dialog modeling and implementation toolkit to explore, prototype, and evaluate [19] a variety of mixed-initiative dialogs. Also, as the need for flexible human-computer dialog in apps for smart phones and Internet-of-Things devices increases, toolkits for improving dialog specification and automating their implementation will find increased use. Lastly, we also envisage the incorporation of natural language, mixed-initiative personal assistants designed and implemented with our toolkit into airport kiosks, ATMs, and interactive, voice-responses systems, since the ubiquity of these platforms in a variety of service-oriented domains, such as education, health care, finance, and travel provide an opportunity for the use of our model for mixed-initiative interaction.

5.2 Future Work

“The advent of virtual, immersive environments in cyberlearning has attracted the attention of researchers [9] and provides a new landscape and opportunity to research models for engineering flexible human-computer dialogs [21]” [26] (see Figure 9). We are currently studying the use of our model in an application, running in an immersive, virtual environment, to support students at a university in their course scheduling activities. The goal of this future work is to study the effect of mixed-initiative dialog through the use of natural language on flexibility in interaction in virtual/cyberlearning environments.

REFERENCES

- [1] J.F. Allen. 1999. Mixed-Initiative Interaction. *IEEE Intelligent Systems* 14, 5 (1999), 14–16.
- [2] T. Ball, C. Colby, P. Danielsen, L.J. Jagadeesan, R. Jagadeesan, K. Läufer, P. Mataga, and K. Rehor. 2000. Sisl: Several Interfaces, Single Logic. *International Journal of Speech Technology* 3, 2 (2000), 93–108.
- [3] M.M. Berg. 2014. *Modelling of Natural Dialogues in the Context of Speech-based Information and Control Systems*. Ph.D. Dissertation. University of Kiel.
- [4] D. Bohus and A.I. Rudnicky. 2003. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proceedings of the Sixth Annual INTERSPEECH Conference*. International Speech Communication Association.
- [5] D. Bohus and A.I. Rudnicky. 2009. The RavenClaw Dialog Management Framework: Architecture and Systems. *Computer Speech and Language* 23, 3 (2009),

- 332–361.
- [6] J.W. Buck and S. Perugini. 2016. A Tool for Staging Mixed-initiative Dialogs. In *Proceedings of the Twenty-seventh Modern Artificial Intelligence and Cognitive Science Conference (MAICS)*, P.H. Phung, J. Shen, and M. Glass (Eds.), 25–32.
- [7] J.W. Buck and S. Perugini. 2017. Mixed-initiative personal assistants. In *Proceedings of the 48th ACM Technical Symposium on Computer Science Education*, T. Barnes and D. Garcia (Eds.). ACM Press, New York, NY, 753–754. Extended abstract.
- [8] A. Dix, J. Finlay, G.D. Abowd, and R. Beale. 2010. *Human-Computer Interaction* (third ed.). Prentice Hall, Harlow, England, Chapter 16: Dialog Notations and Design.
- [9] A. Dubrow. [n. d.]. Seven Cyberlearning Technologies Transforming Education. *Huffington Post*, 6 April 2015. Available: http://www.huffingtonpost.com/aaron-dubrow/7-cyberlearning-technolog_b_6988976.html [Last accessed: 18 November 2015]. ([n. d.]).
- [10] J. Feng, D. Hakkani-Tür, G. Di Fabbrizio, M. Gilbert, and M. Beutnagel. 2006. Webtalk: Towards Automatically Building Spoken Dialog Systems Through Mining Websites. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Computer Society Press, Los Alamitos, CA, 573–576.
- [11] R. Freedman. 2000. Using a Reactive Planner as the Basis for a Dialogue Agent. In *Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference*. 203–208.
- [12] D.P. Friedman, M. Wand, and C.T. Haynes. 2001. *Essentials of Programming Languages* (Second ed.). MIT Press, Cambridge, MA.
- [13] J. Glass and S. Seneff. 2003. Flexible and Personalizable Mixed-initiative Dialogue Systems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (ACL): Human Language Technologies (NAACL-HLT) Workshop on Research Directions in Dialogue Processing*. Association for Computational Linguistics, Stroudsburg, PA, 19–21.
- [14] M. Green. 1986. A Survey of Three Dialogue Models. *ACM Transactions on Graphics* 5, 3 (1986), 244–275.
- [15] C.I. Guinn. 1999. Evaluating Mixed-initiative Dialog. *IEEE Intelligent Systems* 14, 5 (1999), 21–23.
- [16] M.A. Hearst. 1999. Mixed-initiative Interaction. *IEEE Intelligent Systems* 14, 5 (1999), 14.
- [17] J. Hochberg, N. Kambhatla, and S. Roukos. 2002. A Flexible Framework for Developing Mixed-initiative Dialog Systems. In *Proceedings of the Third Association for Computational Linguistics (ACL) SIGDIAL Workshop on Discourse and Dialogue*. Association for Computational Linguistics, Stroudsburg, PA, 60–63.
- [18] N.D. Jones. 1996. An Introduction to Partial Evaluation. *Comput. Surveys* 28, 3 (1996), 480–503.
- [19] P. Jordan, M. Ringenberg, and B. Hall. 2006. Rapidly Developing Dialogue Systems that Support Learning Studies. In *Proceedings of Intelligent Tutoring Systems (ITS) Workshop on Teaching with Robots, Agents, and NLP*. 1–8.
- [20] C. Lee, S. Jung, K. Kim, D. Lee, and G.G. Lee. 2010. Recent approaches to dialog management for spoken dialog systems. *Journal of Computing Science and Engineering* 4, 1 (2010), 1–22.
- [21] A. Leuski and D. Traum. 2011. NPCEditor: Creating Virtual Human Dialogue Using Information Retrieval Techniques. *AI Magazine* 32, 2 (2011), 42–56.
- [22] P. Liang. 2016. Learning Executable Semantic Parsers for Natural Language Understanding. *Commun. ACM* 59, 9 (2016), 68–76.
- [23] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 55–60.
- [24] F. Morbini, E. Forbell, D. DeVault, K. Sagae, D.R. Traum, and A.A. Rizzo. 2012. A mixed-initiative conversational dialogue system for healthcare. In *Proceedings of the Thirteenth Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Stroudsburg, PA, 137–139.
- [25] S. Perugini. 2016. Mining mixed-initiative dialogs. In *Proceedings of the IEEE international conference on systems, man, and cybernetics (SMC)*, S.-F. Su (Ed.). IEEE Computer Society Press, Los Alamitos, CA, 2287–2294.
- [26] S. Perugini and J.W. Buck. 2016. A Language-based Model for Specifying and Staging Mixed-initiative Dialogs. In *Proceedings of the Eighth International ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS)*, José Creissac Campos and A. Schmidt (Eds.). ACM Press, New York, NY, 204–216.
- [27] A. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Stern, K. Lenzo, W. Xu, and A. Oh. 1999. Creating natural dialogs in the Carnegie Mellon Communicator system. In *Proceedings of the Sixth European Conference on Speech Communication and Technology (EUROSPEECH)*. International Speech Communication Association.
- [28] A. Rudnicky and W. Xu. 1999. An agenda-based dialog management architecture for spoken language systems. *IEEE Automatic Speech Recognition and Understanding Workshop* 13, 4 (1999).
- [29] D. Stallard. 2001. Dialogue management in the Talk'n'Travel system. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE Computer Society Press, Los Alamitos, CA, 235–239.