ISSN: 2454-9940

# INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940www.ijsem.org

Vol 12, Issue.3 July 2021

# HELPR: A Cross-Domain Approach to Improving Spoken-Dialog Systems

M.RAMA MOHAN REDDY[1],M.SREEHARI[2],

## Abstract

***People usually interact with intelligent agents (IAs) when they have certaingoals to be accomplished. Sometimes these goals are complex and may requireinteracting with multiple applications, which may focus on different domains. CurrentIAs may be of limited use in such cases and the user needs to directly managethe task at hand. An ideal personal agent would be able to learn, over time, thesetasks spanning different resources. In this paper, we address the problem of crossdomaintask assistance in the context of spoken dialog systems, and describe ourapproach about discovering such tasks and how IAs learn to talk to users about thetask being carried out. Specifically we investigate how to learn user activity patternsin a smartphone environment that span multiple apps and how to incorporate user'sdescriptions about their high-level intents into human-agent interaction***

**Key words**: cross-domain; user intention; spoken dialog systems

## Introduction

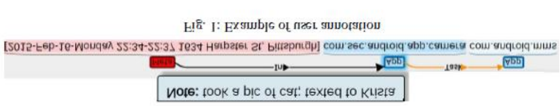
Smart devices, such as smartphones or TVs, allow users to achieve their goals (intentions)through verbal and non-verbal communication. The intention sometimescan be fulfilled in one single domain (i.e., an app). However, the user's intentionis possible to span multiple domains and requires information coordination amongthese domains. A human user, with the global context at hand, can well-organize thefunctionality provided by apps and coordinate information efficiently. On the other hand, although intelligentagents can be configured by developers to passively support(limited) types of cross-domain interactions, they are not capable of activelymanaging apps to satisfy a user's potentially complex intentions, because they donot consider the repeated execution of activities in pursuit of user intentionsCurrently, most human-machine interactions are carried out via touch-screen.Although the vocabularies of recognizable gestures have been expanded during thepast decade [8], interactive expressions are still restricted due to the limit of gesturesand displays. This limit may affect usability, especially for certain populations, suchas older users or users with visual disabilities. By contrast, spoken language can effectivelyconvey the user's high-level and complex intentions to a device. However,the challenges are: 1) understanding both at the level of individual apps and at thelevel of activities that span apps; and 2) communicating a task-level functionalitybetween user and agent. Our previous work focused on predicting user's follow-upaction at app level [25] or understanding the current app-level intention [4]. Thispaper mainly addresses the high-level intention-embedded language understanding.For example, our proposed model understands that "plan a dinner with Alex" is composedof several domains such as YELP, OPENTABLE and MESSENGER.

ASSOCIATE PROFESSOR[1], ASSISTANT PROFESSOR[2],
DEPARTMENT OF ECE
PBR VISVODAYA INSTITUTE OF TECHNOLOGY AND SCIENCE::KAVALI

We alsoenable the system to verbally communicate its understanding of users intentions, inorder to maintain a transparent communication channelMulti-domain dialog systems have been studied in the past [14, 19], where aclassic architecture contains multiple models developed independently for differentdomains and allows corresponding apps to handle user requests [11, 18, 3, 4]. Givena spoken utterance, a domain detector selects 1) a single domain [10, 18, 25, 4] or2) several domains based on the functionality in the user request [20, 21]. However,neither of the two approaches considered the user intention behind the multi-domaininteraction (i.e., why the user needs this set of domains). Our method bridges thelow-level surface forms in cross-domain interactions and the high-level intention inthe user's mind to enable systems to support intention realization. Moreover, consideringa personal assistant's perspective, we compare personalized models withgeneric ones based on personal data availability.The rest of the paper is organized as follows: we first briefly describe a datacollection process to gather user's real-life multi-domain tasks. Then we discuss themethodology to discover, recognized and realize user intentions. Two user studiesare described later as end-to-end and component-wise evaluation.

## Data Collection

We undertook a data collection during which the participants in our study agreedto provide a continuous record of their smartphone use over an extended period oftime, in the form of operating system events (e.g. app invoked, phone number dialed,etc). To do this we implemented an Android app that logs each event, together withits date/time and the phone's location (if GPS is enabled).HELPR: A Framework to Break the Barrier across Domains in Spoken Dialog Systems



Initial analysis of the data indicated that phone usage could be segmented intoepisodes consisting of interaction events closely spaced in time. In our pilot data, wefound 3 mins of inactivity could be used to group events. Although this parameterappeared to vary across users, we used a single value for simplicity. Participantswere asked to upload their log on a daily basis. A privacy step allowed them todelete episodes that they might not wish to share.Due to multi-tasking, episodes might consist of multiple activities, each correspondingto a specific intent. For example one might be communicating with a friendbut at the same time playing a game or surfing the web. We invited participants toour lab on a regular basis (about once a week) to annotate their submitted logs todecouple multiple tasks in the same episodes and also describe the nature (intent) ofthe tasks (see details below). Note that some activities might also span episodes (forexample making plans with others); we did not examine these

## Smartphone Data Annotation

Participants were presented with episodes from their log and asked to group eventsinto sequences corresponding to individual activities [13] (which we will also referto as tasks). Meta-information such as date, time, and street location, was shownto aid recall. Participants were asked to produce two types of annotation, using theBrat server-based tool [23]: 1) Task Structure: link applications that served a commongoal/intention; 2) Task Description: type in a brief description of the goal orintention of the task.
For example, in Fig 1, the user first linked two apps (one about camera and anotherabout text message) together since they were used for the goal of sharing aphoto, and wrote a description "took a pic of ". Some of the task descriptionswere quite detailed and provided the actual app sequence executed (see example inFig 1). However, others were quite abstract, such as "look up math problems" or"schedule a study session". In this paper, we took task descriptions as transcribedintent-embedded user utterances since these descriptions are usually abstract. Weused these descriptions as data for our intention understanding models

4 Ming Sun, Yun-Nung Chen and Alexander I. Rudnicky

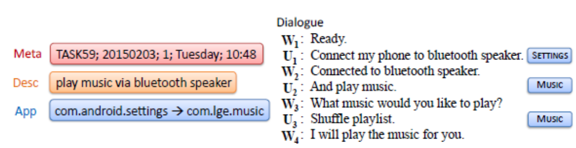



Fig. 2: Multi-app task dialog example. Meta, Desc, App were shown to the participant. Utterances were transcribed manually or via Google ASR. Apps were manually assigned to utterances.

Table 1: Corpus characteristics. Age informally indicates young and old. A native Korean and Spanish speaker participated; both were fluent in English. #Apps is the average number of unique apps. #Multi is the number of tasks which involves multiple user turns.

| Category | #Participants | Age | #Apps | #Tasks | #Multi |
|---|---|---|---|---|---|
| Male | 4 | 23.0 | 19.3 | 170 | 133 |
| Female | 10 | 34.6 | 19.1 | 363 | 322 |
| Age $< 25$ | 6 | 21.2 | 19.7 | 418 | 345 |
| Age $\geq 25$ | 8 | 38.9 | 18.8 | 115 | 110 |
| Native | 12 | 31.8 | 19.3 | 269 | 218 |
| Non-native | 2 | 28.5 | 18.0 | 264 | 237 |
| Overall | 14 | 31.3 | 19.1 | 533 | 455 |

Interactive Dialog Task

We also asked users to talk to aWizard-of-Oz dialog system to reproduce (“reenact”)their multi-domain tasks using speech, instead of the GUI, in a controlled laboratoryenvironment. The users were shown 1) apps used; 2) task description they providedearlier; 3) meta data such as time, location to help them recall the task (see left partin Fig 2). The participants were not required to follow the order of the applicationsused on the smartphones. Other than for remaining on-task, we did not constrainexpression. The wizard (21-year-old male native English speaker) was instructed torespond directly to a participant’s goal-directed requests and to not accept out-ofdomaininputs. An example of a transcribed dialog is shown in Fig 2.This allowed us to create parallel corpora1 of how people would use multipleapps to achieve a goal via both smartphone (touch screen) and language. We recruited14 participants and collected 533 parallel interactions, of which 455 involvemultiple user turns (see Table 1).

HELPR: A Framework to Break the Barrier across Domains in Spoken Dialog Systems 5

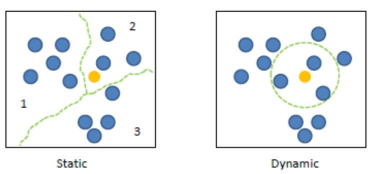


Fig. 3: Illustration of static intention vs. dynamic intention. Blue circles denote training examples and the yellow circle is a testing example.

**Methodology**

For an agent to interact with users at the level of intention, it should 1) understandan intention expressed by speech; and 2) be able to convey its understanding of theintention via natural language. For example, once the user says “I’d like to plana farewell party for my lab-mate”, the agent needs to know the intention behindthis spoken input as well as be able to assist user to find a restaurant (YELP) andschedule time with other lab-mates (MESSENGER). On the other hand, the agentmay reveal its inner state of understanding to the user, especially in clarificationprocess. For instance, it may say “I think we are going to plan an evening event,right?” Channel-maintenance with such verbal cues (either implicit or explicit) ishelpful in conversation [2].We first describe modeling intention understanding, thendescribe the process by which the agent can verbally convey its inner state

**Models for Intention Understanding**

What is user intention? We consider two possibilities. Observed interactions in theintention semantic space may be clustered into KC groups, each representing a specificintention. We refer to this as the static intention. On the other hand, we canalso define dynamic intention, which is a collection of local neighbors (seen interactions)of the input speech. See Fig 3 as an example. In the static intention setting,the agent is aware of the existence of KC intentions and their semantics prior to invocation.However, in the dynamic setting, intention is implicitly defined by the KNnearest neighbors during execution. In both cases, a realization process using themembers of the recognizedintention set maps the user utterance into a sequence/setof apps to support the user activity.We anticipate two major differences between statically and dynamically basedintentions. First, the static approach can use potentially richer information than justintention-embedded utterances when discovering basic intentions — it could usepost-initiate features such as apps launched or user utterances in the spoken dialog.Ideally, this may yield a better semantic space to categorize seen interactions. However, during execution, theinput feature is the same as in the dynamic approach,i.e., task description. Second, the static approach has hard boundaries between intentions.Instances close to the boundaries may not be well characterized by theircluster membersIn both cases the agent will need to map an intention-embedded utterance intosteps (i.e., sequence of apps/domains). Several techniques are available. We cancombine the individual app sequences of the set members into a single app sequencethat represents a common way of surfacing the intention (denoted as REPSEQ). Alternately,we can use a classifier that assigns multiple labels (apps ids) to the input(denoted as MULTLAB). Compared with the MULTLAB strategy, the advantage ofREPSEQ is that it can preserve the order of the app sequence. However, once theintention is classified, the representative app sequence will always be the same, regardlessof variations in the input. This could be a potential problem for staticallybased intentions. Arguably, during this process, we could weight each set memberby its closeness to the input; we did not investigate this possibility. To evaluate, wecompare the set of apps predicted by our realization model with the actual appslaunched by the user and compute an F1 score2.

There are two types of users—ones for which historical data are available, and theothers. New users or users with privacy concerns will not have sufficient data. Thus,a generic model trained from large user community can be used instead of personalizedmodel. We expect that a sufficiently well-trained generic model can providereasonable performance; as history is accumulated performance will improve.

The building of intention understanding models may be impacted by intra- andinter-user inconsistency in the language/apps. We may encounter the problem ofvocabulary-mismatch [13, 22], where interactions related with the same intentionhave non-overlapping 1) spoken terms (words), even caused by minor differencessuch asmisspellings, morphologies, etc; 2) apps, e.g., people may use different apps—MESSENGER or EMAIL with essentially similar functionality. Below we describetwo techniques to overcome potential language- and app-mismatch
Language Mismatch

We can consider a user's input utterances (e.g., "schedule a meeting") as a query tothe intention model. To manage language inconsistency, we used a two-phase process—
1) text normalization where only verbs and nouns in the query are preservedand further lemmatized (e.g., "took"!"take")
2) query enrichment (QryEn) whichexpands the query by incorporating words related to it semantically. QryEn can reducethe likelihood of seeing sparse input feature vector du to out-of-vocabulary [24]words. In this work, we used word2vec [17] with gensim3 toolkit on the pre-trainedGoogleNews word2vec4 model.
The proposed QryEn algorithm is described in Al-

[2] $F_1 = 2 \times Precision \times Recall / (Precision + Recall)$
[3] Toolkit: https://radimrehurek.com/gensim/models/word2vec.html
[4] Model: https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTl:
edit?usp=sharing



gorithm 1. In short, each word $w_i$ in the lemmatized query $Q$ yields mass increases for $N$ semantically close words in the feature vector $f$.

**Algorithm 1** Query Enrichment

**Require:** lemmatized words of the query $Q = \{w_1, ..., w_{|Q|}\}$ and their counts $C = \{c_1, ..., c_{|Q|}\}$; training vocabulary $V$; bag-of-word feature vector $Q_f = \{f_1, ..., f_{|V|}\}$ constructed on $Q$; the word semantic relatedness matrix $M$; the number of semantically similar words $N$ to be extracted for each word in $Q$;
**Ensure:** an enriched bag-of-word feature vector

**for each** $w_i \in Q$ **do**
  Use $M$ to find the $N$ closest words $V_N = \{v_1, ..., v_N\} \in V$;
  **for each** $v_j \in V_N$ **do**
    $f_j = f_j + M_{i,j} \times c_i$
  **end for**
**end for**
**return** $f$;

## App Mismatch

When a generic model is used, recommended apps may not match the apps availableon a specific user's device. For example, the recommended app, BROWSER shouldbe converted to CHROME if that is the only (or preferred) app in this user's phonethat can browse the Internet. Therefore, similarity metrics among apps are needed.

There are several ways to compute app similarity (AppSim). First, based on theedit distance between app (package) names, for example com.lge.music is similarto com.sec.android.app.music since both contains the string "music". Second,we can project an app to a vector space. Ideally, apps with similar functionalitieswill appear close to each other. Possible resources to use are 1) app descriptions inapp stores; 2) language associated with each app when users verbally command theapp (see example in Fig 2). Third, app-store category may indicate functionalitywisesimilarity. However, we found Google Play category too coarse. In this work,we used the first method with 16 fillers (e.g., "android", "com", "htc") removedfrom package names. Examples are shown in Table 2.We found this simple methodsignificantly improved system performance (described later).

Table 2: Most similar apps for Accuweather and Music among 132 apps in our data collection

| order | com.accuweather.android | com.lge.music |
|---|---|---|
| 1 | com.sec.android.widgetapp.ap.hero.accuweather | com.google.android.music |
| 2 | com.jrdcom.weather | com.sec.android.app.music |
| 3 | com.weather.Weather | com.spotify.music |

## Conveying Intention Understanding

IAs may need to communicate with the user in language cast at the level of intention,especially as part of a clarification process. For example, the IA may launch a shortsub-dialog by saying "are you trying to share a picture?" This involves a template("are you trying to ?") and some content ("share a picture"). Instead of echoingcontent directly extracted from the user's current input, we abstract the semantics ofsimilar previous interactions to provide language material indicating that the agent(though a paraphrase) indeed understands the user's intention
**Study**

## Intention Interpretation and Realization

To evaluate intention modeling, we focus on three comparisons: 1) intention:static vs. dynamic models; 2) source: personalized vs. generic setups; 3) method:REPSEQ vs.MULTLAB realization strategies.We used the chronologically first 70%of each user's data for training the personalized model, in principle mirroring actualdata accumulation. The remaining 13 users' first 70% data was combined to train thegeneric model. The number of intentions KC for the static intention model and thenumber of nearest neighbors KN for the dynamic model can be varied. We adaptedKC using gap statistics [26], an unsupervised algorithm, to select the optimal KCfrom 1 to 10 before KMeans. KN was set to the square root of the number of trainingexamples [5]. For REPSEQ we used ROVER to collapse multiple app sequencesinto one [6]. For MULTLAB, we used SVM with linear kernel.

We show system performance in Table 3. This prediction task is difficult since onaverage each user has 19 unique apps and 25 different sequences of apps in our datacollection. The upper part corresponds to static intention model and the lower part todynamic intention. Within either approach, different intention realization strategies(QryEn and AppSim) and their combination are also shown. We performed a balancedANOVA test of F1 score on the factors mentioned above: intention, sourceand method. The test indicates that the performance differs significantly (p<0:05).As noted earlier, the static model has the flexibility to incorporate richer information
(post-initiate features) when used to discover the basic KC intentions. As showin Table 3, adding more post-initiate information (denoted with ? and †) improvepersonalized models since users have behavioral patterns. However, it does not necessarilimprove generic models, mainly due to the inter-user difference in language
and apps.

But we do not observe superior performance for the static model over the dynamicone, even when richer information incorporated (? and †) . For REPSEQstrategy, the dynamic model is much better than the static one. It is possible thatREPSEQ is sensitive to the selection of similar interactions. Arguably, an input mayfall close to the intention boundary in a static setting, which indeed is closer to someinteractions on the other side of the boundary as opposed to the ones within the same

HELPR: A Framework to Break the Barrier across Domains in Spoken Dialog Systems 9Table 3: Weighted average F1 score (%) on test set across 14 participants, using bag-of-words.Average KC in static condition is 7:0_1:0 for generic model, and 7:1_1:6 for personalized model.The static condition was run 10 times and the average is reported. KN in the dynamic conditionis 18:5_0:4 for the generic model and 4:9_1:4 for the personalized model. ? indicates bothdescriptions and user utterances are used in clustering and † indicates apps are used as well.

| | REQSEQ | | MULTLAB | |
|---|---|---|---|---|
| | Personalized | Generic | Personalized | Generic |
| Static (baseline) | 42.8 | 10.1 | 55.7 | 23.8 |
| +QryEn | 44.6 | 11.2 | 56.3 | 27.9 |
| +AppSim | 42.8 | 15.1 | 55.7 | 27.8 |
| +QryEn+AppSim | 44.6 | 16.1 | 56.3 | 36.1 |
| +QryEn+AppSim⋆ | 44.9 | 18.0 | 57.5 | **37.1** |
| +QryEn+AppSim† | **45.8** | **18.1** | **57.6** | 35.9 |
| Dynamic (baseline) | 50.8 | 23.8 | 51.3 | 19.1 |
| +QryEn | 54.9 | 26.2 | 57.0 | 22.9 |
| +AppSim | 50.8 | 30.1 | 51.3 | 22.7 |
| +QryEn+AppSim | **54.9** | **32.5** | **57.0** | **28.0** |

intention cluster. On the other hand, the MULTLAB approach shows relatively consistentperformance in both static and dynamic settings, indicating robustness andself-adaptability with respect to the choice of interactions of similar intention
In Table 3, the fact that QryEn improves the F1 score in all conditions indicatesthat semantic similarity among words can effectively address the languagemismatchproblem. On the other hand, although AppSim has no effect on the personalizedmodel, it addresses the app-mismatch issue in generic models intuitively(p < 0:05 when comparing with the baseline in an balanced ANOVA on additionaltwo factors: intention, method). Combining QryEn and AppSim methods together(denoted as "+QryEn+AppSim") consistently achieves the highest F1 score. As weexpected, generic intention model is consistently inferior to the personalized model.

**Intention Representation in Natural Language**

It should be possible to automatically abstract the semantics of the recognized intentioncluster (or neighbors): Text summarization may be used to generate high-leveldescription of the intention cluster. Keyphrase extraction provides anotheralternative Note that, even if the automatic generation of semantic summarizationis not precise, it may still be sufficiently meaningful in context
In this study, we used the Rapid Automatic Keyword Extraction (RAKE5) algorithm[1], an unsupervised, language-independent and domain-independent extractionmethod. This method has been reported to outperform other unsupervised

Table 4: Mean number of phrases generated using different resources

| MANUAL | ASR | DESC | DESC+ ASR | DESC+ MANUAL |
|---|---|---|---|---|
| 20.0 | 20.3 | 11.3 | **29.6** | 29.1 |

methods such as TextRank [16] and [9] in both precision and F score. In RAKE,we required that 1) each word have 3 or more characters; 2) each phrase have atmost 3 words; and that 3) each key word appear in the text at least once. We didnot investigate tuning these parameters. We use 3 individual

resources and 2 combinations,reflecting constraints on the availability of different contexts in real-life.The three individual resources are manual transcription of user utterances in theirdialogs (MANUAL) and their ASR transcriptions (ASR) and high-level task descriptions(DESC). The average number of key phrases generated by each resource (or
their combination) is shown in Table 4.

We selected 6 users to first review their own clusters, by showing them all clustermembers with 1) apps used in the member interaction; 2) dialog reproduced; 3)meta-data such as time, date, address, etc. We let them judge whether each individualphrase generated by the system summarized all the activities in the cluster (binaryjudgement). We used three Information Retrieval (IR) metrics to evaluate performanceamong different resources — 1) Precision at position K (P@K); 2) MeanAverage Precision6 at position K (MAP@K); 3) Mean Reciprocal Rank (MRR). Thefirst two metrics emphasize on the quality of the top K phrases, MRR focuses on apractical goal — "how deep the user has to go down a ranked list to find one usefulphrase?". Average MRR is 0.64, meaning that the user will find an acceptabledescriptive phrase in the top 2 items shown; an ANOVA did not show significantdifferences between resources. With more sensitive MAP@K and P@K metrics,DESC+ASR and DESC+MANUAL do best. The improvement becomes significantas K increases: having a user-generated task description is very useful.Participants also were asked to suggest carrier phrases that the agent could useto refer to activities; we found these to be unremarkable. Among the 23 phrasescollected, "do you want to " and "would you like to " were the most popular.

To conclude, if the IA can observe a user's speech commands or elicit descriptionsfrom the user (ideally both), it can generate understandable activity referencesand might avoid less efficient interactions (e.g. lists).

## Conclusion and Future Work

We present a framework, HELPR, that is used to learn to understand a user's intentionfrom a high-level description of goals (e.g., "go out with friends") and tolink these to specific functionality available on a smart device. The proposed agentsolicits descriptions from the user. We found that the language used to describe activitiesis sufficient to group together similar activities. Query enrichment and appsimilarity help with language- and domain-mismatch problems, especially when a generic model is used.Wedemonstrated that an agent could use data from large usercommunity while also learning user-specific modelsThe long-term goal of our work is to create agents that observe recurring humanactivities, understand theunderlying intentions and support the task through spokenlanguage interaction. The agent must communicate onthe level of intentions insteadof, or in addition to, individual apps. And it needs to manage the context of theactivity so that its state can be shared between different apps.The value of such an agent is that it would operate on a level higher than providedby app-specific interfaces. It would moreover allow the user to effectively build theirown applications by composing the functionality in existing apps. We have shownthat it is possible to infer user intentions; the next challenge is to capture meaningfulcontext and actively apply it across different apps.

## Acknowledgement

This work was supported in part by YAHOO! InMind, and by the General MotorsAdvanced Technical Center. We thank Zhenhao Hua for implementing the loggerapplication, and YulianTamres-Rudnicky and Arnab Dash for collecting data.

## References

***1. Michael W. Berry and Jacob Kogan. 2010. Text mining: applications and theory.***

***2. Dan Bohus and Alexander I Rudnicky. 2005. Sorry, I didn't catch that!-An investigationof non-understanding errors and recovery strategies. In SIGdial Workshop on Discourse andDialogue (SIGDIAL).***

***3. Yun-Nung Chen and Alexander I. Rudnicky. 2014. Dynamically supporting unexplored domainsin conversational interactions by enriching semantics with neural word embeddings. InProceedings of 2014 IEEE Spoken Language Technology Workshop (SLT). IEEE, 590–595.***
***4. Yun-Nung Chen, Ming Sun, and Alexander I. Rudnicky. 2015. Leveraging Behavioral Patternsof Mobile Applications for Personalized Spoken Language Understanding. In Proceedings of2015 International Conference on Multimodal Interaction (ICMI).***

***5. Richard Duda, Peter Hart, and David Stork. 2012. Pattern Classification. John Wiley andSons.***

***6. Jonathan G Fiscus. 1997. A Post-Processing System to Yield ReducedWord Error Rates: Recognizeroutput voting error reduction (ROVER). In Proceedings of Automatic Speech Recognitionand Understanding Workshop (ASRU). 347–352.***

***7. Kavita Ganesan, ChengxiangZhai, and Jiawei Han. 2010. Opinosis: a graph-based approachto abstractive summarization of highly redundant opinions. In Proceedings of the 23rd***

*internationalconference on computational linguistics (COLING). ACL, 340–348.*

*8. Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E. Hudson. 2014. TouchTools: leveragingfamiliarity and skill with physical tools to augmenttouch interaction. In Proceedings ofthe SIGCHI Conference on Human Factors in Computing Systems. 2913–2916.*

*9. Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge.In Proceedings of the 2003 conference on Empirical methods in natural language processing(EMNLP). ACL, 216–223.*

*10. Qi Li, Gokhan Tur, DilekHakkani-Tur, Xiang Li, Tim Paek, AselaGunawardana, and ChrisQuirk. 2014. Distributed open-domain conversational understanding framework with domain*