Domain Specific Conversational Intelligent Agents: Natural Language Processing in Real World Applications

Mehmet Cagri Calpur[1,2], Mehmet Utku Tatlidede[1], and Irmak Cataloglu[1]

[1] Softtech A.S.
Research and Development Center
Tuzla Piri Reis Cad. 62, 34947 Istanbul, Turkey
[2] Sabanci University
Orta Mahalle, 34956 Tuzla, Istanbul, Turkey
cagri.calpur@softtech.com.tr

Abstract. Natural language processing (NLP) is the branch of Artificial Intelligence (AI) studies that will shape the future of computing and Human Computer Interaction (HCI). Operational complexity of a conversational intelligent agent (Chat-bot) stems from human-related, linguistic and computational aspects. In this study, we define an architectural description of a chat-bot, address the NLP problems which needs to be solved and provide our proposed solutions for a functional conversational intelligent agent. The importance of Bayesian Statistics and Data mining of domain specific text and expected expansion areas are discussed for future research in the conversational AI development.

Keywords: Descriptive Statistics · Machine Learning · Text Data Mining · Ontology · Conditional Random Fields · Conversational Intelligent Agents · Chat-bots · Natural Language Processing.

Alana Özgü Dialog Bazlı Akıllı Ajanlar: Gerçek Dünyada Doğal Dil İşleme Uygulamaları

Mehmet Çağrı Çalpur[1,2], Mehmet Utku Tatlıdede[1], and Irmak Çataloğlu[1]

[1] Softtech A.Ş.
Ar-Ge Merkezi
Tuzla Piri Reis Cad. 62, 34947 İstanbul, Türkiye
[2] Sabancı Üniversitesi
Orta Mahalle, 34956 Tuzla, İstanbul, Türkiye
cagri.calpur@softtech.com.tr

Özet. Doğal dil işleme, yapay zeka çalışmalarının insan bilgisayar etkileşiminin geleceğini şekillendirecek bir alt dalıdır. Sohbet robotu sistemlerinin operasyonel karmaşıklığı insan etkileşiminin çeşitliliğinden kaynaklanan dilbilimsel ve işlemsel zorluklara dayanır. Bu çalışmada, bir sohbet robotu sisteminin mimari tanımını yapıyoruz, alana özgü bir sistemin gerçeklenebilmesi için çözülmesi gereken doğal dil işleme problemlerinden bahsediyoruz ve işlevsel bir sohbet robotu için çözüm önerilerimizi bildiriyoruz. Alana özgü dialog bazlı akıllı ajanların geliştirilmesinde bayesci istatistiksel yöntemlerin ve veri madenciliği tekniklerinin öneminden ve gelecek araştırma alanlarından bahsediyoruz.

Anahtar Kelimeler: Tanımlayıcı İstatistik · Makine Öğrenmesi · Metin Madenciliği · Ontoloji · Şartlı Rastgele Alanlar · Dialog Bazlı Akıllı Ajanlar · Sohbet robotları · Doğal Dil İşleme.

1 Introduction

The imitation of human intuition for language understanding is the motivation behind NLP studies in the industry and academia. Each year new iterations of conversational artificial intelligence agents are entering the Loebner Prize competition. Creation of a generalized intelligent agent which can handle any conversational interaction is infeasible in todays maturity level of NLP research, but a domain specific solution is certainly a reachable target.

Conversational AI (CAI) is replacing current user interfaces and search tools [2,3]. In the coming years, the human computer interaction will be based on text and speech based conversational agents. Such a paradigm shift is limited by the current ability of algorithms powering AI, in this paper we briefly introduce the NLP challenges and in the later sections provide our novel contribution to some of the challenges for our conversational AI platform.

The core of a domain specific conversational AI is a combination of multi-disciplinary research domain. The platform needs a solid software engineering foundation for core components and services. The amount of data and statistical computation requires implementation of efficient algorithms and intuitive design. On the other hand the NLP core is a multi-level process, which starts from acquisition of user input, preprocessing of acquired data, classification and understanding according to the domain concept and producing the response.

2 Natural Language Processing at a Glance

Natural language processing and understanding is one of the more complex contents in AI research. Human beings are naturally inclined to learn sounds, words, patterns and their associations starting from early infancy to their adulthood. The cognitive abilities of a human is not comparable to a machine which needs to be taught how to interpret language. Furthermore this effort of teaching language to machines does not evolve into a thought process for an automaton (machine). The cognitive abilities, intuitive understanding and overwhelmingly fast relational inference capabilities of the human brain enables a human being to understand language even if the speech or text that is being processed is erroneous.

A ***Corpus*** is a large collection of text or speech data, which is used as a basis for NLP studies. ***Corpora*** are usually compiled with manual effort, which can be a dictionary of words, sample sentences or linguistically annotated (morphological decomposition, part-of-speech, named entity information, etc.) text.

2.1 Preprocessing

Natural language understanding (NLU) starts with unstructured data. At the start of the NLU process the text is a combination of alphanumeric and punctuation characters. Rigorous preprocessing is required to normalize the data [8]. The

unstructured data is converted to a standardized format so that NLP algorithms can be developed and the data could be analyzed.

Sentence and Word Decomposition Sentence detection and decomposing a paragraph or unstructured corpus into sentences is important for sequentially processing the corpus to understand the narrative of the text. Although regular structure of a sentence includes punctuation to stop a sentence, fringe cases exist for compound sentences and abbreviations.

Tokenization is separating a sentence into its words with respect to white space characters and punctuation. Decomposing a text into word level is a milestone for applying more advanced NLP algorithms, but a word is not the smallest atomic unit for NLP. A word can be dissected into morphemes.

Lemmatization and Stemming There are two kinds of ***morphemes***. A ***Stem*** is the prime component of a word, which is the base meaning. ***affixes*** are annexed to stems to modify the meaning of the word or form completely different words.

Table 1. Turkish words with different roots may have the same stem string.

Word (Turkish)	**Translation**	**Lemmatization (Root)**	**Stemming**
Boyunluk	Neck collar	Boyun	Boy
Boynu	His neck	Boyun	Boy
Boylar	Tribes	Boy	Boy

Turkish is a member of Altaic language family and a series of suffixes are added to stems to produce words, which is called ***Agglutination***. ***Lemmatization*** is used for determining words that have the same root, despite the mutations and differences in words that originate from the same root. Agglutination is one of the causes of mutation in words, where consonant devoicing occurs when a vowel is appended to the end of a word which ends with an obstruent consonant. ***Stemming*** is a crude form of lemmatization where the stem of the word is extracted but it may not be the actual root of the word under observation. This case is one of the many examples of ambiguity in NLP.

Error Correction and Spell Checking A conversational AI agent's main task is interaction with a human being, therefore the input to the agent is prone to human error. In case of text based communication, there will be misspellings in the form of insertions, deletions, substitutions or transposition of adjacent characters. Damerau [4] claims that %80 of the time the error is one the four error types previously mentioned. Error correction is very important, since errors in the processed text will disrupt the rest of the NLP effort.

2.2 Regular Expressions

Regular expressions (RE) provide a solid tool for completing various NLP tasks. The normalization process becomes much easier and most of the basic NLP tasks can be performed with the help of REs. Unstructured text includes data in various forms and by employing REs, automation of search for recurring data such as numbers, dates or text in special formats becomes much easier. The complexity caused by highly varying data such as numbers can be reduced by substituting numbers with class labels such as *decimal*, *date-US-format*. This kind of substitution is a good way of *feature set reduction* for NLP (Table 2). REs can also be used for simple named entity extraction or part of speech tagging duties.

Table 2. Sample regular expressions.

Regex [1]	**matches**	**Named Entity**
([012][0-9]—3[01])[\/\-\.]([0]*[1-9]—1[012])[\/\-\.][0-9]{4}*	dd.mm.yyyy dd/mm/yyyy dd-mm-yyyy	date-tr-format
[0-9]+([.]{1}[0-9]+)?	10 12.99 0.1	decimal

2.3 Part-of-Speech Tagging

In Turkish there are five types for parts of speech (POS), which are noun, verb, adjective, adverb and conjunction. POS tagging sentences gives us precious information about the words, which is useful for morphological parsing of a word or named entity recognition by examining locations and combinations of POS in a sentence (Table 3). Verbs are generally the terminating POS in a sentence, while the starting POS is usually the subject(noun). Lemmatization and affix determination is useful for POS tagging, since the agglutination property of the language makes conversions such as noun-to-verb, verb-to-noun, etc. possible.

2.4 Named Entity Extraction

A domain specific conversational AI is required to distinguish between regular words and named entities. The scope of named entities changes according to application domain, but usually people, places, organizations, time, date, numerical values, currency can be considered as named entities. The conversation evolves around such special words since named entities are the anchors for context tracking in progressive conversation. Time, space, quantity and related objects and people have attributes and modifiers, so extracted named entities constitute the conceptual existence of the objects in the conversation.

Table 3. POS and NER decomposition of a sample sentence in banking domain.

Sentence	12.06.2018	tarihindeki	kredi kartı	harcamalarımı	göster	.
POS	date-tr-format	adverb	noun	noun	verb	end-of-sentence
NER	date		target-medium	target-type	intent	

2.5 Intent

Intent is the key point or ultimate goal of a sentence in a conversation. A CAI must implement a learning method for determining the intent of a sentence, in order to perform necessary actions and generate appropriate response. Actions and related data are the named entities extracted from the text. The information gathered about the intent is sorted into their respective information slots for completing the required action or generating appropriate response (Table 4).

Table 4. Examples of several intents and their respective named entities that define the required action.

Sentence	**Intent**	**Domain**	**Named entity under focus**
Hello! *Merhaba!*	Greeting	General	-
Show me nearby gas stations. *Yakınlardaki benzin istasyonlarını göster.*	Query	Travel Navigation Transportation	{ “gas station”} { “benzin istasyonları”}
Buy 1000 shares of ABC stock. *1000 adet ABC hissesi satın al.*	Purchase Order	Banking Investment	{“1000 share”, “ABC stock”} {“1000 adet”, “ABC hissesi”}

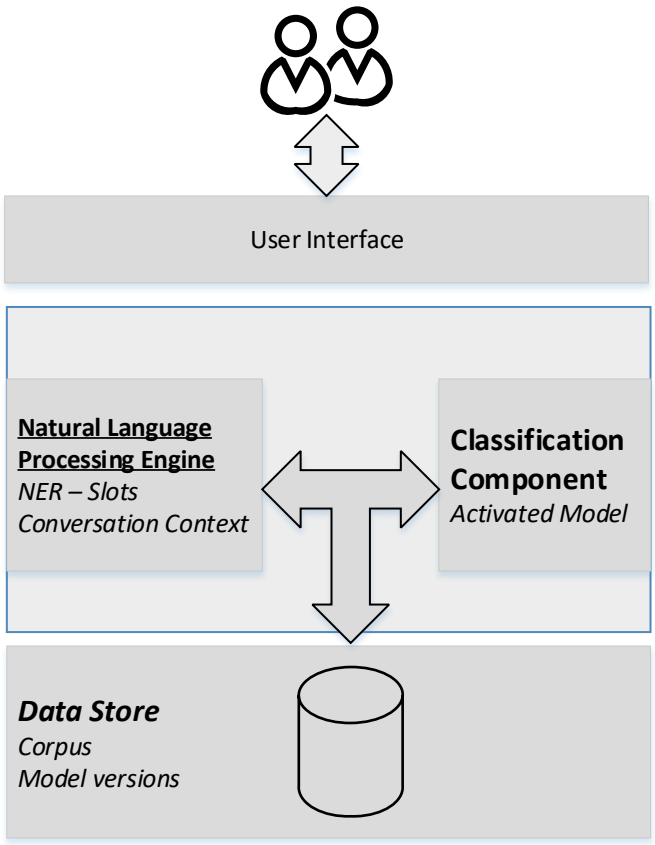
3 Conversational AI Experimental Setup and Results

The domain specific conversational AI project supports the digital and artificial intelligence transition vision of Softtech A.S. The NLP research is implemented as a chat-bot and the solution is deployed at a number of in-house and customer systems as the intelligent information retrieval and query interface.

3.1 Architecture and Components

The CAI system has multi-layered architecture (Figure 1). The *Classification Component* is used for both classification of the input text and training the classification system. Model versioning is implemented and all trained classification models are stored in the data store, so the system adapts to newly available data and roll back to a previous accurate model is possible. Domain specific corpora are also stored in the data store, which is used to initialize the NLP Engine component during start-up of the system. NLP Engine implements various algorithms to provide *Named-Entity-Recognition, Spell Checking* and *Intent Slot Filling.*

Fig. 1. Architecture of the Domain Specific Conversational AI implementation.



3.2 Classification

Multi-Layer Perceptron (MLP) technique is utilized for training and prediction. Training procedure includes 10-fold cross-validation with 80-20 distribution of training and test sets. Configuration of the MLP is 2000 iterations with 0.05 learning rate and hyperbolic tangent activation function. Our test domains included 323 classes and the F1-score for the accuracy is %93 (Table 5).

3.3 Intent and Corpus Generation

A domain specific conversational AI requires expert guidance for the compilation of the classification classes and corpora. The data is usually created by domain experts. Another method is employing market research professionals to gather

Table 5. MLP classification accuracy results for domain specific CAI.

# of classes	323
Accuracy	0.9219
Precision	0.9538
Recall	0.9218
F1 Score	0.9333

data about the domain. For our systems, we use a combination of domain expert generated data and data generated by a market research company. The company owns a mobile crowd sourcing platform and queried 500 people about the domain with this mobile application. Each user prepares 3 sentences for the research. This kind of data generation method could be called the *Wisdom of the Crowds*. Domain experts also checked these results and made necessary modifications and additions.

3.4 Data Mining The Corpus

Data mining effort on the corpus data generates insight about the domain language model. The data can be used to improve the CAI capabilities, such as spell checking. Damerau and Levenshtein algorithms are widely used for tasks in many domains that require a metric to define the difference of sequences of data, such as bioinformatics. But these algorithms are prone to errors [4,7].

Word Window Based Bayesian Spell Checker Our research on application of Bayesian methods on text mining and NLP, led us to an augmentation of the Damerau-Levenshtein word distance metric. All sentences in the domain corpus is processed to generate word windows of a parametric length (Window lengths of 3 & 5 have been tested). A word which will be spell checked stays in the center of the word window. Since any word of a sentence may have errors, the word window requires padding with No-operation (*noop*) labeled null strings for the first and last words of a sentence (Figure 6).

Table 6. Word window combinations for a sample sentence.

Sentence	*Yakınlardaki benzin istasyonlarını göster*				
Word Windows size = 5	Noop	Noop	yakınlardaki	benzin	istasyonlarını
	Noop	yakınlardaki	benzin	istasyonlarını	göster
	yakınlardaki	benzin	istasyonlarını	göster	Noop
	benzin	istasyonlarını	göster	Noop	Noop

The word window is a combination of specific words, each window has an occurrence frequency. Normalized-Levenshtein similarity is calculated from the Turkish dictionary for the misspelled word and the metric is reinforced with

the frequency of matching word windows for the misspelled word. Word windows are generated from an input text with misspelled word and the pattern is searched in the collection of all word windows generated from the corpus. The middle word of the word window with the highest *(frequency * similarity)* multiplication value is suggested as the correction. Our proposed algorithm (1) incorporates domain specific similarity metric, therefore the spell checking process becomes domain context aware. Context awareness reduces the faults related to Damerau similarity metric, where equal scores are generated for words with exactly identical errors with the same character length; i.e. $(incorrectword, suggestion1, suggestion2) \leftarrow (xool, tool, cool)$.

Algorithm 1: Misspelling correction based on Domain Specific Bayesian Methods

Data: Domain Corpus, Dictionary
Result: Bayesian Word Window (WW) Based Correct Word Suggestion
C = Word Window Collection with frequencies from Corpus;
foreach $word \notin Dictionary$ **do**
  $S \leftarrow Damerau - Levenshtein(word)$;
  if $word = middleword \leftarrow WWpatternmatch\ in\ C$ **then**
    $PairSet < word, frequency > \leftarrow S * WWfrequency$;
  return $word \leftarrow MaxFrequency(PairSet < word, frequency >)$;

4 Conclusion and Future Work

In this paper we have given brief description of some of the NLP topics that require immediate solution for implementing a *Domain Specific Conversational AI*. Also we have discussed our approach to development of such a system and suggested an improved algorithm to a fundamental problem that affects whole NLP process.

A Domain specific conversational AI is a miniaturized sub-problem of generalized conversational intelligence. Deep learning techniques are taking over all areas of AI and machine learning research, but the curse of dimensionality [5] is an issue for the size of the problem at hand. Application of Bayesian methods to this problem looks promising for areas such as unsupervised morphological inference of affixes [6] or text summarization problem [9].

References

1. Debuggex: Online visual regex tester. javascript, python, and pcre. https://www.debuggex.com/, accessed:2018-06-17
2. Insight artificial intelligence ui. https://www.accenture.com/us-en/insight-artificial-intelligence-ui, accessed: 2018-06-17
3. Top trends in the gartner hype cycle for emerging technologies, 2017. https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/, accessed: 2018-06-17

4. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Commun. ACM **7**(3), 171–176 (Mar 1964). https://doi.org/10.1145/363958.363994, http://doi.acm.org/10.1145/363958.363994
5. Donoho, D.L.: High-dimensional data analysis: The curses and blessings of dimensionality. In: AMS CONFERENCE ON MATH CHALLENGES OF THE 21ST CENTURY (2000)
6. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Comput. Linguist. **27**(2), 153–198 (Jun 2001). https://doi.org/10.1162/089120101750300490, https://doi.org/10.1162/089120101750300490
7. Greenhill, S.J.: Levenshtein distances fail to identify language relationships accurately. Comput. Linguist. **37**(4), 689–698 (2011)
8. Jurafsky, D., Martin, J.H.: Speech and Language Processing (2Nd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2009)
9. Takamura, H., Okumura, M.: Learning to generate summary as structured output. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. pp. 1437–1440. CIKM '10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1871437.1871641, http://doi.acm.org/10.1145/1871437.1871641