In-the-wild chatbot corpus: from opinion analysis to interaction problem detection

Irina Maslowski[1,2], *Delphine Lagarde*[1], *Chloé Clavel*[2]

[1]EDF Lab Paris-Saclay, Palaiseau, France
[2]Télécom ParisTech, Paris, France

irina.maslowski@edf.fr, delphine.lagarde@edf.fr, chloe.clavel@telecom-paristech.fr

Abstract

The past few years have seen growing interests in the development of online virtual assistants. In this paper, we present a system built on chatbot data corresponding to conversations between customers and a virtual assistant provided by a French energy supplier company. We aim at detecting in this data the expressions of user's opinions that are linked to interaction problems. The collected data contain a lot of "in-the-wild" features such as ungrammatical constructions and misspelling. The detection system relies on a hybrid approach mixing hand-crafted linguistic rules and unsupervised representation learning approaches. It takes advantage of the dialogue history and tackles the challenging issue of the opinion detection in "in-the-wild" conversational data. We show that the use of unsupervised representation learning approaches allows us to noticeably improve the performance (F-score = 74.3%) compared to the sole use of hand-crafted linguistic rules (F-score = 67,7%).

Index Terms: Chatbot dialog, Interaction problem, Opinion mining, Human-computer interaction, Written interactions

1. Introduction

Virtual agents and chatbots taking the role of on-line advisers have recently gained in popularity in the websites of the companies. The challenge remains the same as for human advisers: to improve customer satisfaction. In this paper, we propose to contribute to the detection of problematic interactions in a written chat with a virtual adviser with a system named DAPI [1]. The present study takes place in the concrete application context of a French energy supplier EDF, using EDF chatbot corpus, gathering "in-the-wild" and rich spontaneous expressions.

We rely on the definition of [1] which defines a problematic situation as a reflection of the user's dissatisfaction with the conversational system answer. We call such kind of situations *interaction problems* (IP). We propose a hybrid approach to detect IP: hand-crafted linguistic rules based on finite state transduction over annotations and unsupervised representation learning to determine the word semantics.

Hitherto, the majority of studies that have tried to predict or to detect problems in human-machine interactions, were carried out for spoken dialog systems (SDS). Various types of cues are thus used to detect IP: prosodic cues [2, 3], speech-based system logs [4] – such as the low confidence of the outputs of the speech recognition, the direct feedback of the users about their satisfaction towards the interaction [5], semantic [3] and linguistic cues [6, 3, 7]. The studies carried out on chat-oriented dialog systems are still less numerous, even though the use of chatbot systems by companies is increasing.

Linguistic features for the detection of IP are classically used as an input of supervised machine learning techniques. They range from basic linguistic features such as bag of words, n-grams [6, 3] and basic linguistic distances [3, 8] to Parts of Speech (POS) and statistic term frequency-inverse document frequency features [9]. The linguistic cues integrate various context of the dialogue history ranging from one to six user - agent turns [6, 3].

Some approaches integrate scores of semantic similarity between utterances in order to detect IP. For example, [3] use the inner product to calculate the score of semantic similarity between sentence vectors. The sentence vectors are build using neural network approaches. [1] use a knowledge-base for the same task for a general domain chat.

Other studies choose to also use opinion or affect cues in order to detect IP : [3], (in a SDS) and [1] (in a general domain chatbot in Chinese) use a lexicon-based approach for the detection of the affect or a sentiment in order to detect a problematic communication. [1] enhance a lexicon-based approach by regular expressions to model sentiment patterns.

In line with [1]'s approach on a general domain online chatbot in Chinese language, we propose a pioneering study that considers the user's opinions and emotions for the detection of IP in a domain-specific chatbot (customer relationship for electricity company) in French. Our main contributions are as follows: i) to take advantage of the entire dialogue history: the rules integrate linguistic cues contained in all preceding user's utterances; ii) to model the IP as the expressions of user spontaneous opinion or emotion towards the interaction; iii) to integrate web-chat and in-the-wild language specificities as linguistic cues for our rules; iv) to take advantage of word embeddings representations learned on our big unlabelled chatbot corpus in order to model semantic similarities.

In the following, we present our corpus of human-virtual agent written dialogues (Section 2). We introduce our system architecture (Section 3) and discuss our system evaluation results (Section 4). Finally we conclude and speak about our future work directions (Section 5).

2. Human - Virtual Agent Chat Corpus

The corpus (described in detail in[10]) contains all the interactions between users and the virtual agent (VA) collected from EDF company web-site from January to November 2014 totaling 1,813,934 dialogues. The role of the VA is to answer the users' questions about the EDF website navigation or the services and products of the company. A dialogue is composed at

[1] "Détection Automatique de Problèmes d'Interaction" (automatic detection of interaction problems)

least of one adjacent pair (AP) that contains a user's utterance and a VA utterance.
The dialogues contain "Failed" metadata given by the chatbot system but we are not using those as to remain as generic as possible in our corpus usage. The corpus of the EDF company has been anonymized and is private.

The main feature of the corpus is that it carries characteristics of French chat as described by [11]: emoticons (though rare), abbreviations, a phonetic spelling, "echo characters", multiple punctuation and Anglicisms. The corpus contains typing and misspelling errors: 12% of words are tagged as `<unknown>` by TreeTagger [12] [2]. This specificity of a gathered "in-the-wild" corpus renders the data difficult to process. However, such linguistic specific features are important because they carry information on the user opinion or emotions [10], e.g. "Parfait merci ;)" or "pfff".

A subset of the big corpus was annotated in IP using GATE interface [14]. Following the strategy presented in [15] in order to simplify the annotation task, we define an annotation process guided by questions and information summaries. An IP taxonomy was thus proposed (see Figure 1) and integrated within a simplified decision tree. The taxonomy allows distinguishing *explicit interaction problems (EIP)* (an expression of the user negative emotion or opinion towards the interaction) and *implicit interaction problems (IIP)* (other linguistic clues: user's repetitions, user's contact request or "how does it work?" inquiries). The *EIP* are represented by a relation consisting of a triplet: source - opinion - target. The representation of a user's opinion or a user's emotion is based on the relation model from the appraisal theory of [16]. We have chosen this model according to the analysis of existing approaches exposed in [17]. We will use *OPEM* acronym which stands for OPinion and EMotion in order to gather all the opinion-related phenomena. Only the OPEM that have the interaction as a target were annotated. The interaction as the target, can be mentioned by the user *explicitly* (e.g. "tu es virtuelle, tu ne peux pas m'aider" [you are virtual, you can not help me]) or *implicitly* (e.g. Agent: "Veuillez m'excuser, je n'ai pas compris ce que vous venez de dire." [I beg your pardon, I have not got what you said.] User: "pffff").

We have held two manual annotation campaigns to create: i) the "DevCorpus", for the development of the current system; ii) the "T-Corpus" for the evaluation of the current system. We choose to call upon a specialist in semiology – familiar with the analysis of the corpora of the EDF company – for the annotation. Even though this choice does not allow us to obtain a quantitative measure of the annotation reliability, it corresponds to a good compromise between reliability and annotation cost. The corpora statistics are presented in Table 1 and shows that both corpora contain a similar proportion of IP. In both corpora, the ratio of dialogues with at least one IP is relatively low: 25% and 28% respectively. Only 15,5% of all user's utterances contain an IP. Only 11% of IP in the development corpus and 6% of IP in the reference corpus are explicit. IP are annotated at the utterance level. Despite the fact that our system does not need to detect the fine classes of IP, they are a good support for the linguistic analysis of the system annotation results.

Table 1: *Statistics of manually annotated corpora.*

Main Statistics	DevCorpus	T-Corpus
Dialogues	3,000	3,000
Adjacent Pairs (AP)	8,576	8,630
Dialogues with at least one IP	741	845
AP with IP	15%	17%
Problematic AP in a problematic dialogue (mean)	2	1.5

3. Hybrid Approach

The DAPI system aims to detect utterances containing IP in written conversations between a user and a VA by analyzing in real-time the user's utterances. The overall architecture of our system based on the GATE framework [14] is presented in Fig. 2. It relies on a hybrid approach combining hand-crafted linguistic rules and unsupervised representation learning to determine the word semantics. After a preprocessing step, the linguistic rules are used in order to extract expressions of user's negative opinion towards the interaction and other linguistic cues of IP. They rely on the GATE JAPE (Java Annotation Patterns Engine) that provides finite state transduction over annotations [18] based on regular expressions. The linguistic rules take advantage of dialogue history and integrate Internet French chat features. The learned word semantics is used to improve the detection of user's repetitions and problem reformulations that are featuring IP (Section 3.3.3).

The preprocessing is composed of the data anonymization, the elimination of hyper-links, the text tokenization, and the POS and chunks annotation by TreeTagger [12]. According to the used version of DAPI (see Section 4), it is possible to include a spell checking step using PyEnchant[3] library of Python. In order to avoid cleaning valuable clues of IP, before applying the spell checker, we verify that words are not in the dictionaries of Internet slang[4] (e.g. *lol*), emotions (lists of emotions and insults from LIWC for French [19]) and business terms (lexicon of business terms grouped into concepts and consisting of 400 entries provided by the EDF company and constructed on the basis of different business corpora including our *DevCorpus*). This preliminary check is carried out using "difflib" library[5] of Python, which is an extension of the Ratcliff and Obershelp algorithm [20]. We describe the following processing steps according to the type of context which is taken into consideration.

3.1. At the level of the user's utterance

The annotation rules are designed to detect relations between a source, an OPEM, and a target. They combine lexical clues based on Internet slang, LIWC and several small hand-made lexicons of: basic emoticons, potential sources of opinion (first personal pronoun variants, as we focus on the user's opinion), opinion verbs and expressions (20 entries), expressions of different concepts (*e.g.* gratitude, greetings, demand) (30 entries). *Relations* are modelled by seven relation patterns $[Source\ OPEM\ Target]$ depending on the presence of a target, a source and an OPEM in the same sentence. First, each element (source, OPEM or target) that can potentially be a part of a relation is detected. Then, if matching a relation pattern, the user's utterance is annotated as containing an IP. The *potential OPEM* is modelled by thirteen rules of three and four levels of complexity. They include the negation processing which is

[2] It's worth noting that, a similar assessment was done in the customer - human agent chat corpus presented in [13]

[3] `http://pythonhosted.org/pyenchant/`

[4] `http://fr.wiktionary.org/wiki/Annexe:Liste_de_termes_d%E2%80%99argot_Internet`

[5] `https://docs.python.org/3/library/difflib.html#difflib.SequenceMatcher`

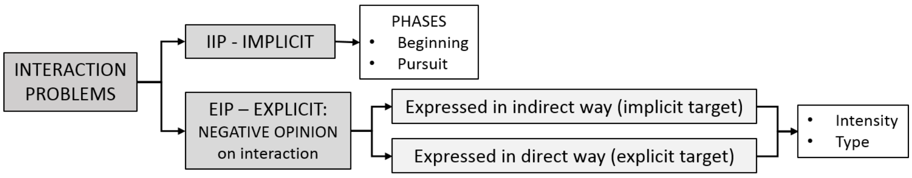
INTERACTION PROBLEMS
IIP - IMPLICIT
PHASES
• Beginning
• Pursuit
EIP – EXPLICIT: NEGATIVE OPINION on interaction
Expressed in indirect way (implicit target)
Expressed in direct way (explicit target)
• Intensity
• Type

Figure 1: *Taxonomy of Interaction Problems*

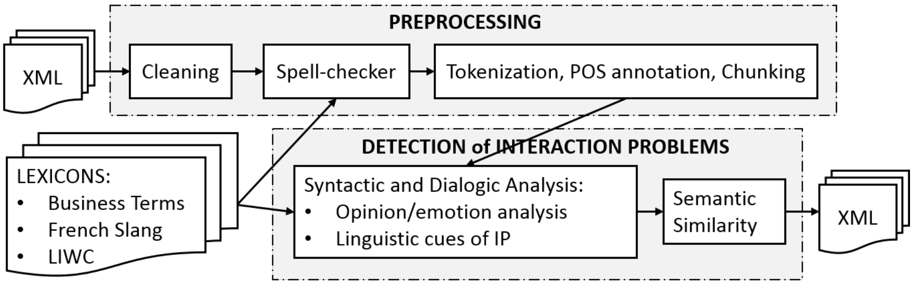
PREPROCESSING
XML
Cleaning
Spell-checker
Tokenization, POS annotation, Chunking
DETECTION of INTERACTION PROBLEMS
LEXICONS:
• Business Terms
• French Slang
• LIWC
Syntactic and Dialogic Analysis:
• Opinion/emotion analysis
• Linguistic cues of IP
Semantic Similarity
XML

Figure 2: *DAPI System Scheme*

based on [15] chunk approach. Typographical clues such as the multiple punctuation and the smileys, and expressions of dissatisfaction (ex. ”Laisse moi tranquille”) [leave me alone] are used to detect relations with an implicit target.

The *explicit interaction target* is modeled by eleven sub-concepts linked to: the mutual comprehension during the interaction, the efficiency of the VA's work or the adequacy of the VA's answer. The sub-concept ”réponse”, for example, contains the following list of synonyms: ”réponse, résultat, réaction, solution, explication, réplique, retour”[6].

3.2. Using the context of agent's utterance

Spontaneous Contact Requests. We define a spontenous user contact request as a set of lemmas of the following groups of words: 1) ”contacter” [to contact], ”téléphoner” [to phone], ”téléphone” [a phone]; 2) conseiller [advisor], EDF [company name], where at least one word of each group should be present in the user's utterance, otherwise a string ”appel” [a call] should be present. The rules based on the detection of the *user contact requests* are the following: 1) contact requests (like in [9]) and inquiries on the chatbot functioning are treated as problematic if they are not in the first user's utterance; 2) if the Agent's contact suggestion comes before the user's request, the user's utterance is not considered as problematic.

Expressions of dissatisfaction towards agent's answer. We use expressions of user dissatisfaction to detect IP according to the following rule:
IF *the agent expresses its inability to help the user*
AND *the user's utterance contains echo characters and/or onomatopoeia (”pfff”) or Internet slang expressions as means to close the dialogue (ex. English word ”bye”),*
THEN *the user's utterance is labeled* `<interaction problem>`.

3.3. Modelling over several user's turns : user repetition and reformulation

We use the following approaches to detect the *user repetitions or reformulations*: linguistic distances, the detection of the repetition of business concepts or terms and semantic similarity.

3.3.1. Detecting user's repetitions by using linguistic distances.

We calculate linguistic distances between the current user's utterance and all the preceding user's utterances by applying the Jaccard distance improved by [21] and the Levenshtein distance [22]. The Jaccard distance measures the common part of the vocabularies for two user's utterances. The Levenshtein distance measures the differences between the character sequences in order to manage typing errors. It is worth noting that, though these distances are the most commonly used metrics for user repetition detection [3, 8], the Jaccard distance we use allows a better performance in long phrases. The final distance for an utterance is the minimal distance between the current utterance and each previous utterance. The rules are based on the comparison of the distances to thresholds ($\leqslant 4$ for Levenshtein and $\leqslant 0.85$ for Jaccard) that have been optimized on the *DevCorpus* in order to detect IP. The Levenshtein distance is complementary to the Jaccard distance as it detects user repetitions containing misspelled words.

3.3.2. Detecting user's repetitions by retrieving business concepts.

Two different rules are based on business concepts[7] or terms. The retrieving of business concepts is carried out with the lexicon of business concepts and with patterns dedicated to retrieve multi-word expressions of business terms such as *'customer space'*. The first rule is based on multiple punctuation and constant *presence of business terms* in user's utterances. It also takes into account the dialogue history. The presence of

[6] answer, result, reaction, solution, explanation, reply, feedback

[7] a business concept is a synset of business terms

business terms in the previous user's utterances disambiguate multiple punctuation concerning the interaction from that concerning products or services. The rule is as follows:
IF *a user's utterance contains a business term followed by a multiple punctuation (ex. !!, ??)*,
AND *a business term was already contained in the previous user's utterances*,
THEN *the user's utterance is labeled* `<interaction problem>`.

In the second rule, we are looking for the presence of the same business concept in the previous user's utterances. If a business concept on the current user's utterance was already mentioned in one of the previous user's utterances, the current utterance is annotated as containing an IP. This is the case of the third user's utterance (U3) in Example 1. In this example, the business terms "carte bleue" [credit card] and "carte bancaire" [bank card] belong to the same concept "Carte Bancaire".

Example 1 *Detection of an interaction problem based on the repetition of business concepts*
User [U1]: *je régler par carte bleu je ne le trouve plus*[8]
Agent: *EDF met plusieurs [URL] modes de paiement à votre disposition. (...)*[9]
User [U2]: *[URL]*
Agent: *Je viens de vous rediriger vers la page demandée.*[10]
User [U3]: *je veut régler par carte bancaire*[11]
Agent: *(...)*

3.3.3. Detecting user's reformulation by using semantic similarity measures and word embeddings.

We use the *semantic similarity* in order to detect more user reformulations (DAPI-3 and DAPI-4). The computation of the semantic similarity between two user's utterances is based on the representation of words in a vector space. We have allocated the larger part of our corpus (named the "ChatBot Embedded" corpus) for training the word/utterance embeddings models. The "ChatBot Embedded" raw corpus contains 2,112,860 user's utterances (11,087,419 words). The corpus went through the following transformations: the separation of articles from words, the letter case homogenization, deletion of numbers, nonce words and stop words. The final number of words is 8,888,049. We have chosen the **word2vec model** [23] to transform the words of our corpus into vectors. We use the standard word2vec library for python [12] with the following training parameters: size = 100, cbow = 0, verbose = False, iter = 5. The word vectors are summed to obtain the vector of the utterance. The cosine distance between the vectors of two utterances with a threshold of 0.85 (optimized on the *DevCorpus*) determines whether two utterances are similar. If so, the second user's utterance is annotated as an IP. The following section presents the results of the evaluation of DAPI system.

4. Evaluation and Discussion

To our knowledge, there is no other system that can serve us as a baseline for the detection of IP in a French written chat with a virtual adviser. Hereafter, we describe the steps we follow to establish a baseline.
As the major clues of IP are the users' repetitions/reformulations and the users' opinions/emotions on the interaction, we apply two methods separately: the classique Jaccard distance [24] to detect repetitions and the Naïve Bayes classification which is commonly used for sentiment analysis [25]. The 0.15 threshold for the Jaccard distance is determined on the basis of the best trade-off between recall and precision on the DevCorpus. The 10-fold cross-validation is applied to the Naïve Bayes classification on the T-Corpus. Considering the low results of the both approaches (see Table 2), we choose the basic configuration of our system (DAPI-1) as a baseline.
In order to evaluate the contributions of the spell checker and of the word embeddings representation, we compare four versions of our system: **DAPI-1** system with only linguistic rules; **DAPI-2** integrating the spell-checker, **DAPI-3** integrating the computation of the score of semantic similarity but not the spell-checker and **DAPI-4** combining both the spell-checker and the computation of the score of semantic similarity. The systems are evaluated on the *T-corpus* by computing Precision, Recall and F-score [26] for the detection at the utterance level. The IP detection scores are shown in Table 2.

Table 2: *Results in % for the detection of IP in the T-corpus.*

System	Precision	Recall	F-score	Accuracy
Naïve Bayes	25.9	14.6	18.6	90.1
Jaccard	55.5	38.6	45.6	79.2
DAPI-1	72.4	63.6	67.7	90.1
DAPI-2	72.0	65.4	68.5	90.2
DAPI-3	72.0	77.0	74.4	91.4
DAPI-4	71.1	**77.8**	74.3	91.3

The use of word embeddings (DAPI-3 and DAPI-4) provides a noticeable improvement of the system performance. DAPI-3 obtains the best F-score. However, DAPI-4 allows a higher recall, which is important in our context (it is important to detect the maximum of existing IP). It is worth noting that we have also experimented to train word2vec on the corpus processed with the spell-checker but the results of the calculation of the score of semantic similarity dropped. The utterances detected as similar using the semantic similarity can be characterized as: user repetitions with a highly misspelled context (rules using linguistic distances detect simpler cases of repetitions); reformulations containing words with the same word-root (e.g. the word "payer"[13] in the user's utterance " je ne trouve pas ma facture pour la *payer* en ligne"[14] and "paiement"[15] in "je ne veux pas le télépaiement je veux le *paiement* par carte bleu"[16], similarity score 0.877) and reformulations containing at least one expression in common (e.g. the expression "je souhaite"[17] in the following user's utterances "bonjour, *je souhaite* voir le récapitulatif de mes prélèvement[18]"/ "*je souhaite* savoir combien je suis relevé par mois[19]", similarity score 0.869).

The linguistic rules based on the tracking of the repetition of business concepts detect reformulations as well. These are reformulations containing business terms with a common root

[8] I pay with a credit card I can not find it any more

[9] EDF puts several payment methods at your disposal.(...)

[10] I have just redirected you to the requested page.

[11] I want to pay by bank card

[12] `https://pypi.python.org/pypi/word2vec`

[13] to pay

[14] I can not find my bill to pay it online

[15] payment

[16] I don't want the telebanking, I want the credit card payment

[17] I would like

[18] Goodday, I would like to see the summary of my withdrawals

[19] I would like to know how much is my bank withdrawal per month

(e.g. "pourquoi paie t on d'avance l'abonnement"[20]/ "paiement abonnement d'avance"[21], where the words with the common root are "payer"[22] and "payement"[23]). The joint use of both approaches to the detection of the user reformulation as a mark of IP contributes to the robustness of the system to cope with the challenges of the "in-the-wild" corpus. However, both our approaches to the user reformulation detection (business concept repetition and semantic similarity) still create a lot of false positives (e.g. in the cases when the user clarifies his/her previous utterance or carries on with the same topic) that are difficult to handle.

The joint model of the specificities of the chat language and the dialogue history contributes, for example, to detecting a user irritation towards the interaction with the chatbot (the rule combining multiple punctuation and business terms). In particular, multiple punctuation clues take an important role in the detection (78,5% of correct matches done with the rules exploiting the specificities of the chat language, are done considering the multiple punctuation clue).

5. Conclusion and Future Work

In this paper, we present the DAPI system based on a hybrid approach for the detection of interaction problems in dialogues between a human and a virtual adviser. The system focuses on the expressions of user spontaneous opinion or emotion that feature interaction problems. DAPI combines an approach based on hand-crafted rules for finite state transduction over annotations and semantic similarity measures computed on word embeddings learnt from a big unsupervised corpus. We have tried different configurations of DAPI system. The best performance from the application point of view (higher recall) is obtained by the version of the system combining the semantic similarity and the linguistic rules with the spell-checker. The semantic similarity based on word embeddings detects complex user reformulations and misspelled repetitions. In future work, we would like to investigate other types of hybridization between unsupervised representation learning and rule-based approaches, allowing to take advantage of our *big* unlabeled chatbot corpus.

6. References

[1] Y. Xiang, Y. Zhang, X. Zhou, X. Wang, and Y. Qin, "Problematic situation analysis and automatic recognition for chinese online conversational system," *Proc. CLP*, pp. 43–51, 2014.

[2] A. Batliner, C. Hacker, S. Steidl, E. Nöth, and J. Haas, "User states, user strategies, and system performance: how to match the one with the other," in *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, 2003.

[3] S. Georgiladakis, G. Athanasopoulou, R. Meena, J. Lopes, A. Chorianopoulou, E. Palogiannidi, E. Iosif, G. Skantze, and A. Potamianos, "Root cause analysis of miscommunication hotspots in spoken dialogue systems." in *INTERSPEECH*, 2016, pp. 1156–1160.

[4] M. A. Walker, I. Langkilde-Geary, H. Wright Hastie, J. Wright, and A. Gorin, "Automatically training a problematic dialogue predictor for a spoken dialogue system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 293–319, 2002.

[5] H. W. Hastie, R. Prasad, and M. Walker, "What's the trouble: automatically identifying problematic dialogues in darpa communicator dialogue systems," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 384–391.

[6] A. van den Bosch, E. Krahmer, and M. Swerts, "Detecting problematic turns in human-machine interactions: Rule-induction versus memory-based learning approaches," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2001, pp. 82–89.

[7] F. Cailliau and A. Cavet, "Mining automatic speech transcripts for the retrieval of problematic calls," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2013, pp. 83–95.

[8] J. Liscombe, G. Riccardi, and D. Hakkani-Tür, "Using context to improve emotion detection in spoken dialog systems," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[9] I. Beaver and C. Freeman, "Detection of user escalation in human-computer interactions." in *INTERSPEECH*, 2016, pp. 2075–2079.

[10] I. Maslowski, "Quelles sont les caractéristiques des interactions problématiques entre des utilisateurs et un conseiller virtuel?" *PARIS Inalco du 4 au 8 juillet 2016*, p. 94, 2016.

[11] F. Achille, "Constitution d'un corpus de français tchaté," in *RECITAL*, Dourdan, France, 2005.

[12] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of International Conference on New Methods in Language Processing*. Manchester, UK: Association for Computational Linguistics, 1994.

[13] A. Nasr, G. Damnati, A. Guerraz, and F. Bechet, "Syntactic parsing of chat language in contact center conversation corpus," in *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016, p. 175.

[14] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters, *Text Processing with GATE (Version 6)*, 2011. [Online]. Available: http://tinyurl.com/gatebook

[15] C. Langlet and C. Clavel, "Improving social relationships in face-to-face human-agent interactions: when the agent wants to know user s likes and dislikes," in *ACL 2015*, 2015.

[16] J. R. Martin and P. R. White, "The language of evaluation," *Appraisal in English. Basingstoke & New York: Pal grave Macmillan*, 2005.

[17] C. Clavel and Z. Callejas, "Sentiment analysis: from opinion mining to human-agent interaction," *IEEE Transactions on affective computing*, vol. 7, no. 1, pp. 74–93, 2016.

[18] H. Cunningham, D. Maynard, and V. Tablan, "Jape-a java annotation patterns engine , department of computer science, university of sheffield," 2000.

[19] A. Piolat, R. Booth, C. Chung, M. Davids, and J. Pennebaker, "The french dictionary for liwc: Modalities of construction and examples of use— la version franaise du dictionnaire pour le liwc: modalités de construction et exemples d'utilisation," 2011.

[20] J. W. Ratcliff and D. E. Metzener, "Pattern-matching-the gestalt approach," *Dr Dobbs Journal*, vol. 13, no. 7, p. 46, 1988.

[21] E. Brunet, "Peut-on mesurer la distance entre deux textes?" *Corpus*, no. 2, 2003.

[22] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[24] P. Jaccard, "Distribution de la flore alpine dans le bassin des drouces et dans quelques regions voisines." vol. 37(140), pp. 241–272, 1901.

[20] why do we pay in advance the subscription

[21] the in advance payment of subscription

[22] to pay

[23] the payment

[25] F. Saad, “Baseline evaluation: an empirical study of the performance of machine learning algorithms in short snippet sentiment analysis,” in *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*. ACM, 2014, p. 6.

[26] C. Van Rijsbergen, “Information retrieval. dept. of computer science, university of glasgow,” *URL: citeseer. ist. psu. edu/vanrijsbergen79information. html*, vol. 14, 1979.