Information Retrieval Chatbots Based on Conceptual Models

Tatiana Makhalova[1,2(✉)], Dmitry Ilvovsky[1], and Boris Galitsky[3]

[1] National Research University Higher School of Economics, Moscow, Russia
{tpmakhalova,dilvovsky}@hse.ru
[2] Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
[3] Oracle Corp., Redwood Shores, CA, USA
boris.galitsky@oracle.com

Abstract. Customer support systems based on chatbots gain an increasing popularity. Chatbots are becoming more and more important to a plethora of applications not only for social services. Modern information retrieval (IR) chatbots are based on simple queries to a database and do not ensure intelligent dialogues with users. In this paper we propose an IR-chatbot model that incorporates a concept-based knowledge model and an index-guided traversal through it to ensure the discovery of information relevant for users and coherent to their preferences. The proposed approach not only supports a search session, but also helps users to discover properties of items and sequentially refine an imprecise query.

Keywords: Formal Concept Analysis · Pattern Structures · Chatbots

1 Introduction

Dialogue systems and conversational agents – including chatbots, personal assistants and voice-control interfaces – are becoming ubiquitous in modern society. However, building intelligent conversational agents remains a major unsolved problem in artificial intelligence research.

There are two main directions in the chatbot development.

(1) *"Social chatbots"* are able to keep any kind of conversation, they usually do not follow a perfectly-defined trajectory and might be spontaneous. The development of the social chatbots implies accumulating a huge set of training dialogue data and feeding it to a deep learning network. The chatbot is expected to learn automatically "how to chat" [3].

(2) *"Task-oriented chatbots"* are expected to keep a natural-style conversation in a particular domain, e.g., restaurant information retrieval, booking a flight, providing automatic customer support [10,11]. They are developed with the help of a wide range of NLP and ML functionality. Domain-specific knowledge is incorporated in the model via explicitly-provided features and model-output

D. Endres et al. (Eds.): ICCS 2019, LNAI 11530, pp. 230–238, 2019.
https://doi.org/10.1007/978-3-030-23182-8_17

Table 1. A catalog fragment (a). Objects g_1–g_6 are smartwatches, g_7 is a dress and g_8 is a sweatshirt. The corresponding "is defined" formal context is given in (b), see details in Sect. 3.

	Brand	Avg. review	Price	Heart rate	GPS	Waterproof	Color	Material	Size
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
g_1	TrainYS	3.6	105	yes	yes	yes	Black, Purple, White, Silver		
g_2	Youroach	4.8	180	yes	yes	no	Black, Pink, White, Silver		
g_3	BestWatch	2.1	50	no	yes	no	Black, White, Red		
g_4	Youroach	4.5	265	yes	yes	yes	Black, Red, White, Silver, Blue		
g_5	TrainYS	3.8	15	no	no	yes	Black		
g_6	CDream	4.8	250	yes	yes	yes	Black, White, Blue		
g_7	Brand X	3.9	140				White, Red, Violet	silk	XS
g_8	Brand Z	2.4	85				Green, Red	wool	S,M,L

(a)

	Brand	Avg. review	Price	Heart rate	GPS	Waterproof	Color	Material	Size
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
g_1	×	×	×	×	×	×	×		
g_2	×	×	×	×	×	×	×		
g_3	×	×	×	×	×	×	×		
g_4	×	×	×	×	×	×	×		
g_5	×	×	×	×	×	×	×		
g_6	×	×	×	×	×	×	×		
g_7	×	×	×				×	×	×
g_8	×	×	×				×	×	×

(b)

restrictions [15], partially observable Markov decision processes [16] and other techniques [2,6].

Being unreliable and too brittle, these two approaches are unsuitable for enterprise chatbots.

In the epoch of web search engines, there is a need for a specific class of task-oriented chatbots – those who support web search in case of imprecise queries in specific domains, they are called information retrieval(IR)-chatbots. Their essential features are (i) ability to give a relevant and short list of items even when the user has only a general idea of the item he/she is searching for; (ii) flexibility, i.e., the user may change his/her mind about some preferences being provided with information from a chatbot, the chatbot should update a search strategy w.r.t. the user answers and (meta)data; (iii) efficiency, i.e., the user should get a satisfiable result within a short communication.

The existing IR-chatbots send queries to database to support web search and do not use any domain knowledge models. In this paper we propose a conceptual model of chatbots that is based on a knowledge model traversal rather than simply queries to database. We use Formal Concept Analysis (FCA) and Pattern Structures (PS) to build a knowledge model and stability index (its approximation) to ensure an efficient query refinement procedure (i.e., to satisfy user needs within a short communication). The paper is organized as follows. Below we provide a small example where IR-chatbots might be used. In Sect. 2 we present the basic notions of FCA and PS. Section 3 introduces a knowledge model that is the base of IR-chatbots. In Sect. 4 we discuss principles for the IR-chatbot functioning and present an algorithm for the interactive refinement of user queries. In Sect. 5 we conclude and give direction of future work.

Use Case. Let us consider a small use case where the IR-chatbots can be used. Given a catalog of objects (data) that are organized in categories (metadata) the user goal is to find a particular object(s) in the catalog. A fragment of a catalog and the corresponding hierarchy is given in Table 1(a) and Fig. 1, respectively.

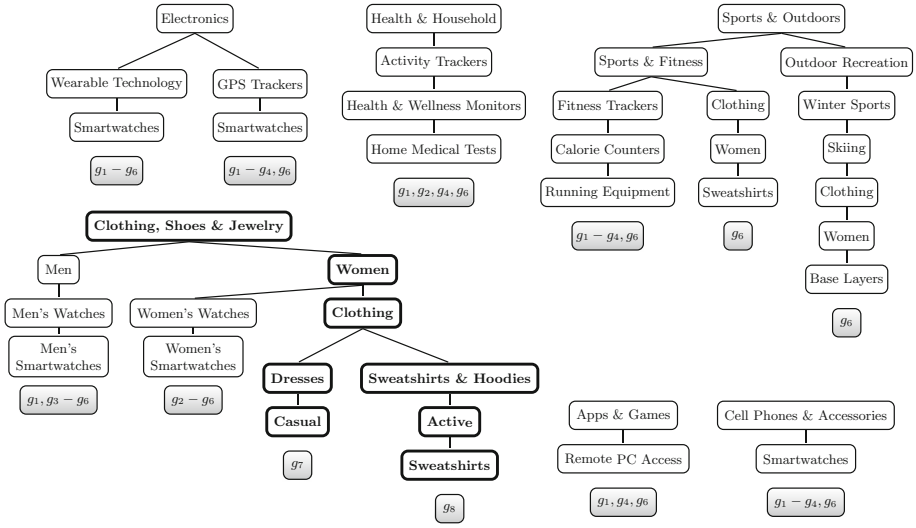

Fig. 1. A fragment of a hierarchy of categories for objects from Table 1. The category names for objects g_7 and g_8 are highlighted in bold.

Usually, objects are categorized ambiguously, i.e., are located in several leafs of a hierarchical tree. An object might belong to completely different or repetitive categories. That might lead users astray rather than improves search experience. For example, searching for "smartwatch" one can get a long list of categories (see paths with the leafs containing g_1–g_6 in Fig. 1). It is unclear beforehand if some items can be found in different categories. More than that, a web search engine usually proposes irrelevant attributes to refine and/or does not adjust user preferences w.r.t. available items.

Thus, the main problem is that in case of very general queries the search engine does not provide convenient tools to discover a variety of objects the user might be interested in.

In the next section we present a theoretical framework that allows us to remove ambiguity in metadata and improve the user experience by taking into account user-specified constraints and the variety of values of object satisfying given constraints.

2 Basic Notions

A *formal context* [8] is a triple (G, M, I), where $G = \{g_1, g_2, ..., g_n\}$ is a set objects, $M = \{m_1, m_2, \ldots, m_k\}$ is a set attributes and $I \subseteq G \times M$ is an incidence relation, i.e. $(g, m) \in I$ if the object g has the attribute m. The derivation operators $(\cdot)'$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$A' = \{m \in M \mid \forall g \in A : gIm\}\,,\ B' = \{g \in G \mid \forall m \in B : gIm\}\,.$$

A' is the set of attributes common to all objects of A and B' is the set of objects sharing all attributes of B. A *(formal) concept* is a pair (A, B), where $A \subseteq G$, $B \subseteq M$ and $A' = B$, $B' = A$. A is the *extent* and B is the *intent* of the concept. A partial order $\leqslant$ is defined on the set of concepts as follows: $(A, B) \leqslant (C, D)$ iff $A \subseteq C\,(D \subseteq B)$, (A, B) is a subconcept of (C, D), (C, D) is a superconcept of (A, B).

Pattern Structure [9] is a generalization of the formal context, where objects are described by more complex structures. A *pattern structure* is a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a complete meet-semilattice of descriptions and $\delta : G \to D$ is a mapping of an object to a description. The Galois connection between a set of objects and their descriptions is defined as follows: $A^{\square} := \sqcap_{g \in A} \delta(g)$ for $A \subseteq G$, $d^{\square} := \{g \in G \mid d \sqsubseteq \delta(g)\}$ for $d \in D$. A pair (A, d) for which $A^{\square} = d$ and $d^{\square} = A$ is a *pattern concept*.

Intensional stability [12,13] of concept (A, B) is the probability that B will remain closed when removing a subset of objects from extent A with equal probability: $stab((A, B)) = |\{C \subseteq A \mid C' = B\}|/2^{|A|}$. The concepts with high values of stability are more stable w.r.t. random removal of the objects.

The computing stability is $P\#$-complete [12]. In practice, one uses its approximations [1,4]. One of the most popular approximations is Δ-measure [5]. That is defined as the minimal difference in supports between concept (A, B) and its nearest subconcepts, i.e., $\Delta((A, B)) = \min_{(A^*, B^*) \leqslant (A, B)} |A| - |A^*|$.

3 Building a Domain Knowledge Model

We propose to use a two-level knowledge model, where the upper level is used to navigate through groups of similar objects described in a very general way and the bottom level is used to search particular objects within a group of similar objects described in detail.

Upper Level: Coarse Categorization Model. Let G be a set of objects described by a set of attributes M, where every attribute $m \in M$ has a particular domain of values $dom(m)$. We define a binary relation I for $g \in G$ and $m \in M$ as follows: $gIm = 1 \Leftrightarrow g$ has m. We call I "is defined" relation, see example in Table 1.

We define *degree of homogeneity* of objects $A \subseteq G$ as the rate of their common attributes, i.e. $h(A) = |\{\{g\}' \mid g \in A\}|/|M|$, homogeneity is anti-monotonic. Partially ordered concepts (A, B) having $h(A) \in [h_1, h_2]$ make the upper level $\mathcal{M}_{\mathcal{U}}$. The lower bound h_1 prevents from creating too general groups of objects, while the upper bound h_2 ensures that very homogeneous (similar in sense of attributes M, not their values) objects will be treated in detail at the bottom level. For every concept (A, B) we add the names of categories of objects from A, these names are extracted from metadata, we denote it by $F(A)$. To build this set we need to collect all category names of each $g \in A$ from the roots to leafs in a given hierarchy (metadata). For example, the homogeneity rate of $(\{g_7, g_8\}'', \{g_7, g_8\}')$ is 6/9. The category names to be collected for this concept are highlighted in bold in Fig. 1.

Bottom Level: Refined Categorization Model. For each set of objects A, where $(A, B) \in \mathcal{M}_{\mathcal{U}}$, we build a pattern structure $\mathcal{M}_{\mathcal{B}}(A)$ using the corresponding data fragment of the original dataset. The derivation operator $^{\square}$ is defined specifically for different attribute types.

4 Interactive Query Refinement

4.1 Two-Stage Approach: Basic Idea

We propose an approach to query refinement where a user query is specified in an interactive manner. The procedure is adaptive w.r.t. both user constraints and the variability of attributes of objects satisfying the constraints.

The interactive query refinement consists of two main steps:

1. Navigating to a relevant concept $(A, B) \in \mathcal{M}_{\mathcal{U}}$.
2. Query refinement within $\mathcal{M}_{\mathcal{B}}(A)$.

The pseudocode of the procedure is given in Algorithm 1. For a given user query Q in natural language, we use standard NLP methods to extract a set of key words K and values V of some attributes from M (line 1). Then the concepts $(A, B) \in \mathcal{M}_{\mathcal{U}}$ for which $F(A)$ matches with K (line 2) are chosen as most relevant concepts $\mathcal{F}$ and ranked w.r.t. Δ-measure. The iterative refinement starts from the top-ranked concept. At each iteration in lines 5–8 the algorithm tries to find the items relevant to the user among objects from A (line 7).

Input: Q, a user query in natural language; $\mathcal{M}_{\mathcal{U}}$, the upper level of the knowledge model
Output: A^*, a subset of objects that meet user needs

1 $(K, V) \leftarrow ExtractWordsAndValues(Q)$
2 $\mathcal{F} \leftarrow$ NavigateToConcept$(\mathcal{M}_{\mathcal{U}}, K)$ ▷ see Algorithm 2
3 $\mathcal{F}_{ord} \leftarrow sortByDelta(\mathcal{F})$
4 $A^* \leftarrow \emptyset$
5 **while** $A^* = \emptyset$ **do**
6 $\quad A \leftarrow next(\mathcal{F}_{ord})$
7 $\quad A^* \leftarrow$ refineQuery$(\mathcal{M}_{\mathcal{B}}(A), V)$ ▷ see Algorithm 3
8 **end**

Algorithm 1. Two-stage query refinement

4.2 The First Stage: Navigating to a Group of Relevant Homogeneous Objects

To find most relevant formal concepts we traverse $\mathcal{M}_{\mathcal{U}}$ starting from most general concepts, see Algorithm 2. We compare the keywords K from the query with $F(A)$, $(A, B) \in \mathcal{M}_{\mathcal{U}}$ (line 3). At each iteration we try to get more specific groups of objects from the set of lower neighbors of $(A, B) \in \mathcal{M}_{\mathcal{U}}$, i.e., $LowerNeighbors(\mathcal{M}_{\mathcal{U}}, A)$. We stop specifying concepts when no more specific concept (A_s, B_s) of (A, B) $(A_s \subset A)$ has the same number of matched keywords as (A, B), i.e. when $|K \cap F(A)| > |K \cap F(A_s)|$ (condition in line 3 is false for all $A \in \mathcal{F}$).

4.3 The Second Stage: Walk Through a Pattern Structure

Once $(A, B) \in \mathcal{M}_{\mathcal{U}}$ has been identified, the corresponding pattern structure $\mathcal{M}_{\mathcal{B}}(A)$ is chosen to support further query refinement, see Algorithm 3.

The idea of the refinement is a sequential questioning of the user, where a new user response is used to jump to the most promising pattern concept (A_r, d_r), d_r is analyzed by the $isVaried$ function to identify the features that will be offered the user to refine.

On input the algorithm gets a pattern structure $\mathcal{M}_{\mathcal{B}}(A)$ and, optionally, values V of some attributes from M extracted from the query. We start from the most specific concept (A_s, d_s) whose description includes values from V (line 2). If V is empty, the most general (top) pattern concept is taken. Its lower neighbors $LowerNeighbors(A_s, d_s)$ ordered by Δ-measure and the number of lower neighbors. They are candidates for further refinement. The attributes M^* to be specified are chosen with help of the $isVaried$ function (line 7). Boolean function $isVaried$ checks if values for given objects are quite diverse to be specified by the user. Examples of $isVaried$ function are given in Table 2, (a). M_{irrel} are attributes marked by the user as irrelevant.

Once attributes $M^* \subseteq M$ are identified (line 7) the chatbot asks the user to choose attributes $M^*_{spec} \subseteq M^*$ he/she wants to specify. Then the chatbot shows the user only particular values of M^*_{spec} depending on d_r instead of the whole rage of values $dom(\cdot)$. That facilitates the decision-making process. The specified values W_{spec} are used to find the next (A_s, d_s).

Example. Let us illustrate the principles of the query refinement using the running example given in Table 1. For query "smartwatch" the chatbot navigates to pattern structure $\mathcal{M}_{\mathcal{B}}(\{g_1, \ldots, g_6\})$. Since no attribute values V have been extracted we start from the top concept. The sequences of refinements is given in Fig. 2.

All lower neighbors of the top concept have Δ-measure equal to 1, thus we start from the pattern concept with extent $A_1 = \{g_1, g_2, g_3, g_4, g_6\}$ since it has the smallest number of lower neighbors. The attributes that meet $isVaried$ requirements (see Table 2) are *"Price"* and *"Waterproof"*. The user chooses *"Waterproof"* and the chatbot finds a concept with extent $A_2 = \{g_1, g_4, g_6\}$ that satisfies user preferences. Based on $isVaried$ condition, the chatbot proposes to choose *"Avg. review"* or *"Price"* for refinement. The user choses highly

Input: K, a set of key words (category names); $\mathcal{A}$, a set of most general concepts
Output: $\mathcal{F}$, a set of concepts to be specified

1 $\mathcal{F} \leftarrow \mathcal{A}$
2 **for** $A \in \mathcal{F}$ **do**
3 **if** $K \subseteq F(A)$ **then**
4 $\mathcal{F}^* \leftarrow LowerNeighbors(\mathcal{M}_{\mathcal{U}}, A)$
5 $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}^*$; $\mathcal{F} \leftarrow \mathcal{F} \setminus \{A\}$
6 **end**
7 **end**

Algorithm 2. NavigateToConcept

Input: $\mathcal{M}_{\mathcal{B}}(A)$, a pattern structure; V, attribute values extracted from the query
Output: *relevant*, a set of relevant objects

```
run ← True; relevant ← ∅; M_irrel ← ∅
(A_s, d_s) ← findTheMostSpecificConcept(M_B(A), V)
GS_ranked ← sortByDelta(LowerNeighbors(A_s, d_s))
while run and GS_ranked ≠ ∅ do
    (A_r, d_r) ← pop(GS_ranked)
    relevant ← A_r
    M* = {M_i | M_i ∈ M \ M_irrel, M_i ∈ isVaried(A_r, M)}
    if M* ≠ ∅ then
        M*_spec = ask(M*)                    ▷ ask the user for categories he/she wants to specify
        M_irrel ← M* \ M*_spec
        S ← {b | b ∈ values(M_i, d_r), M_i ∈ M*_spec}    ▷ suggest particular values within chosen
         categories
        W_spec ← userChoice(S)
        if W_spec ≠ ∅ then
            (A_s, d_s) ← findTheMostGeneralConcept(A_r, d_r, W_spec)
            GS_ranked ← sortByDelta(LowerNeighbors(A_s, d_s))
        end
    end
end
return relevant
```

Algorithm 3. REFINEQUERY

rated items, i.e., the range $[4.0, 4.8]$ and get items g_4 and g_6 that are too expensive for him/her. The search refinement continues with next candidate $A_3 = \{g_1, g_2, g_3, g_5, g_6\}$. Then the objects are $A_4 = \{g_1, g_2, g_6\}$, the relevant item is g_1.

Table 2. The condition for $isVaried$ boolean function (a) and the attributes that satisfy $isVaried$ condition (highlighted in bold) and are offered the user to refine (b).

Condition on variability	**Description**
$\mathcal{E}_1 : v_1 < \frac{\lvert m'(g) \cap A \rvert}{\lvert A \rvert} < v_2$	rate of objects having m
$\mathcal{E}_2 : v_1 < \frac{\lvert A \rvert}{\lvert \{m(g) \mid g \in A\} \rvert} < v_2$	relative number of different values for nominal attr.
$\mathcal{E}_3 : std([m(g) \mid g \in A]) > v$	standard dev., for real-valued attr.

(a)

Attribute	$isVaried$	A_1	A_2	A_3	A_4
Brand	$0.5 \leq \mathcal{E}_1 \leq 0.6$	0.8	1	0.8	1
Avg. review	$0.5 \leq \mathcal{E}_4 \leq 1.13$	1.15	**0.62**	**1.11**	**0.69**
Price	$50 \leq \mathcal{E}_3 \leq 100$	**92.4**	**88.36**	**95.7**	**72.5**
Heart rate	$0.5 \leq \mathcal{E}_1 \leq 0.6$	0.8	1	**0.6**	1
GPS	$0.5 \leq \mathcal{E}_1 \leq 0.6$	1	1	**0.6**	1
Waterproof	$0.5 \leq \mathcal{E}_1 \leq 0.6$	**0.6**	1	**0.6**	0.66
Color	$2 \leq \mathcal{E}_2 \leq 5$	7	6	7	6

(b)

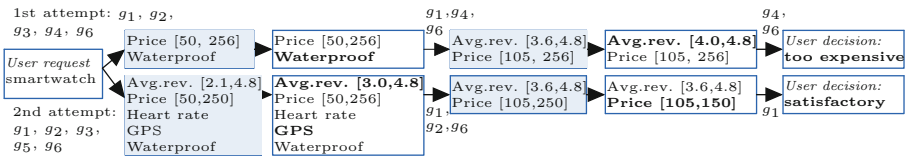

Fig. 2. A sequence of query refinements. Colored rectangles correspond to chatbot proposals, the attributes in bold in white rectangles are user choices. The sets of objects out of rectangles refer to the concepts chosen by the chatbot according to the user-specified preferences.

5 Conclusion

In this paper we introduce a new model of the IR-chatbot. To the best of our knowledge it is the first model that is based on traversal through a knowledge model rather than simple queries to database. We use Δ-measure to obtain satisfactory search results in a small number of steps. The proposed model has the following advantages w.r.t. traditional IR-chatbots: (i) it is adaptive to both user preferences and the range of items that satisfy user criterion; (ii) it helps users to discover relevant properties of the items he/she is searching for based on Δ-measure and $isVaried$ function; (iii) it offers the user a small number of attributes to refine at each step, that simplifies the decision-making process.

The presented model can be incorporated in the developing project on discourse structure-driven dialog system and semantic-based web search tools [7,14].

Acknowledgements. Sections 3, 4.1 and 4.2 (algorithm to build a domain knowledge model, algorithm to navigate to a group of relevant objects) were written by Dmitry A. Ilvovsky and Tatiana Makhalova supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Russia.

Sections 2 and 4.3 (chatbot investigations, model and algorithm of pattern structure walk) were prepared within the framework of the HSE University Basic Research Program and funded by the Russian Academic Excellence Project '5–100'. The rest of the paper was written and performed at Oracle Corp.

References

1. Babin, M.A., Kuznetsov, S.O.: Approximating concept stability. In: Domenach, F., Ignatov, D.I., Poelmans, J. (eds.) ICFCA 2012. LNCS (LNAI), vol. 7278, pp. 7–15. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29892-9_7
2. Bordes, A., Boureau, Y.L., Weston, J.: Learning end-to-end goal-oriented dialog. arXiv preprint arXiv:1605.07683 (2016)
3. Bowden, K.K., Oraby, S., Misra, A., Wu, J., Lukin, S., Walker, M.: Data-driven dialogue systems for social agents. In: Eskenazi, M., Devillers, L., Mariani, J. (eds.) Advanced Social Interaction with Agents. LNEE, vol. 510, pp. 53–56. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-92108-2_6
4. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Scalable estimates of concept stability. In: Glodeanu, C.V., Kaytoue, M., Sacarea, C. (eds.) ICFCA 2014. LNCS (LNAI), vol. 8478, pp. 157–172. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07248-7_12
5. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Sofia: how to make FCA polynomial? In: Proceedings of FCA4AI, vol. 1430, pp. 27–34 (2015)
6. Eric, M., Manning, C.D.: Key-value retrieval networks for task-oriented dialogue. arXiv preprint arXiv:1705.05414 (2017)
7. Galitsky, B.: Semantic tools. https://github.com/bgalitsky/relevance-based-on-parse-trees
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-642-59830-2

9. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS-ConceptStruct 2001. LNCS (LNAI), vol. 2120, pp. 129–142. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44583-8_10
10. Henderson, M., Thomson, B., Williams, J.D.: The second dialog state tracking challenge. In: Proceedings of SIGDIAL, pp. 263–272 (2014)
11. Hirschman, L.: Evaluating Spoken Language Interaction: Experiences from the Darpa Spoken Language Program 1990–1995. Spoken Language Discourse. MIT Press, Cambridge (2000)
12. Kuznetsov, S.O.: On stability of a formal concept. Ann. Math. Artif. Intell. **49**(1–4), 101–115 (2007)
13. Kuznetsov, S., Obiedkov, S., Roth, C.: Reducing the representation complexity of lattice-based taxonomies. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS-ConceptStruct 2007. LNCS (LNAI), vol. 4604, pp. 241–254. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73681-3_18
14. Makhalova, T., Ilvovsky, D., Galitsky, B.: News clustering approach based on discourse text structure. In: Proceedings of the First Workshop on Computing News Storylines, pp. 16–20 (2015)
15. Williams, J.D., Asadi, K., Zweig, G.: Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. arXiv preprint arXiv:1702.03274 (2017)
16. Young, S., Gašić, M., Thomson, B., Williams, J.D.: POMDP-based statistical spoken dialog systems: a review. Proc. IEEE **101**(5), 1160–1179 (2013)