Chapter 12
A Social Promotion Chatbot

Abstract We describe a chatbot performing advertising and social promotion (CASP) to assist in automation of managing friends and other social network contacts. This agent employs a domain-independent natural language relevance technique that filters web mining results to support a conversation with friends and other network members. This technique relies on learning parse trees and parse thickets (sets of parse trees) of paragraphs of text such as Facebook postings. To yield a web mining query from a sequence of previous postings by human agents discussing a topic, we develop a Lattice Querying algorithm which automatically adjusts the optimal level of query generality. We also propose an algorithm for CASP to make a translation into multiple languages plausible as well as a method to merge web mined textual chunks. We evaluate the relevance features, overall robustness and trust of CASP in a number of domains, acting on behalf of the author of this Chapter in his Facebook account in 2014–2016. Although some Facebook friends did not like CASP postings and even unfriended the host, overall social promotion results are positive as long as relevance, style and rhetorical appropriateness is properly maintained.

12.1 Introduction

A conventional chatbot is designed as a communication means between a customer and a company. In this section we propose a totally different area of a chatbot activity: social promotion. We design a chatbot that communicates with peers on behalf of its human host. Instead of answering questions about products and services, or fulfilling requests from the users, this social chatbot is representing its human host in maintaining relationships with her friends and colleagues. The goal of this chatbot is to relieve its human host from routine activity of casual conversation with peers. Also, as an additional feature, this chatbot can implicitly advertise products and services, mentioning them in a conversation with human peers as long as it fits the context of such conversation (Galitsky 1998).

Simulated human characters are increasingly common components of user interfaces of applications such as interactive tutoring environments, eldercare systems,

B. Galitsky, *Developing Enterprise Chatbots*,
https://doi.org/10.1007/978-3-030-04299-8_12

virtual reality-based systems, intelligent assistant systems, physician-patient communication training, and entertainment applications including (Cassell et al. 2000; De Rosis et al. 2003; Dias and Paiva 2005; Lisetti 2008; Galitsky and Parnis 2017; Trias et al. 2010) among others. While these systems have improved in their intelligent features, expressiveness, understanding human language, dialog abilities and other aspects, their social realism is still far behind. It has been shown (Reeves and Nass 1996) that users consistently respond to computers as if they were social actors; however, most systems do not behave as competent social actors, leading to user loose of trust and alienation.

Most users used to distrust conversational agent who has shown poor understanding of their needs in the areas such as shopping, finance, travel, navigation, customer support and conflict resolution (Galitsky et al. 2005). To restore trust in chatbots, they have to demonstrate robust intelligence features on one hand and operate in a domain where users are more tolerant to agent's misunderstanding of what chatbots say or recommend (Galitsky and McKenna 2017).

In this section we build a chatbot in the form of simulated human character that acts on behalf of its human host to relieve her from the routine, less important activities on social networks such as sharing news, and commenting on postings of others. Unlike the majority of application domains for simulated human characters, its social partners do not necessarily know that they deal with an automated agent. We refer to this character as a *chatbot* that assists its human host [possibly, with *advertising*] and *social promotion* (CASP). Over the years, we experimented with CASP in a number of Facebook accounts (Galitsky et al. 2014) and evaluated its performance and trust by human users communicating with it.

To be trusted, a chatbot operating in a natural language must produce relevant content in an appropriate situation and suitable target person. To do that, it needs to implement the following intelligence features (Lawless et al. 2013):

1. Flexibility in respect to various forms of human behavior, information sharing and request by humans;
2. Resourcefulness, being capable of finding relevant content in an emergent and uncertain situation;
3. Creativity in finding content and adjusting existing content to the needs of human user;
4. Real-time responsiveness and long-term reflection on how its postings being perceived;
5. Use of a variety of reasoning approaches, in particular based on simulation of human mental states;
6. Ability to learn and adapt performance at a level of intelligence seen in humans and animals;
7. Awareness of and competence in larger natural, built, and social contexts.

For a chatbot, users need to feel that it properly reacts to their actions, and that what it replied makes sense. To achieve this in a limited, vertical domain, most effective approaches rely on domain-specific ontologies. In a horizontal domain, one needs to leverage linguistic information to a full degree (Sidorov et al. 2012;

Fig. 12.1 Dimensions of social promotion

Galitsky et al. 2012) to be able to exchange messages in a meaningful manner. Once we do not limit the domain a chatbot is performing in, the only available information is language syntax and discourse (Strok et al. 2014), which should be taken into account by means of a full scale linguistic relevance filter.

Social promotion (Fig. 12.1) is based on

1. involvement (living the social web, understanding it, going beyond creation of Google+ account);
2. creating (making relevant content for communities of interest);
3. discussing (each piece of content must be accompanied by a discussion. If an agent creates the content the market needs and have given it away freely, then you will also want to be available to facilitate the ensuing discussions);
4. promoting (the agent needs to actively, respectfully, promote the content into the networks).

CASP acts in the environments subject to constant changes. As news come, political events happen, new technologies are developed and new discoveries are made, CASP needs to be able to find relevant information using new terms or new meanings of familiar terms and entities (Makhalova et al. 2015). Hence it needs to automatically acquire knowledge from the web, expanding its taxonomy of entities and building links between them (Chap. 8, Galitsky 2013). These taxonomies are essential when CASP needs to match a portion of text found on the web (as a candidate message) against a message posted by a human user. By growing these taxonomies, CASP learns from the web, adapts its messages to how the available information on the web is evolving (Galitsky and Ilvovsky 2016). Also, CASP applies accumulated the experience from user responses to its previously posted messages to new posting.

Paragraphs of text as queries appear in the search-based recommendation domains (Montaner et al. 2003; Bhasker and Srikumar 2010; Galitsky 2017) and social search (Trias and de la Rosa 2011). Recommendation agents track user chats, user postings on blogs and forums, user comments on shopping sites, and suggest web documents and their snippets, relevant to a purchase decisions (Galitsky and Kovalerchuk 2006). To do that, these recommendation agents need to take portions of text, produce a search engine query, run it against a search engine API such as Bing or Yahoo, and filter out the search results which are determined to be irrelevant to a purchase decision. The last step is critical for a sensible functionality of a recommendation agent, and a poor relevance would lead to a problem with retaining users.

12.2 The Domain of Social Promotion

On average, people have 500–800 friends or contacts on social network systems such Facebook and LinkedIn. To maintain active relationships with this high number of friends, a few hours per week is required to read what they post and comment on it. In reality, people only maintain relationship with 10–20 most close friends, family and colleagues, and the rest of friends are being communicated with very rarely. These not so close friends feel that the social network relationship has been abandoned.

However, maintaining active relationships with all members of social network is beneficial for many aspects of life, from work-related to personal. Users of social network are expected to show to their friends that they are interested in them, care about them, and therefore react to events in their lives, responding to messages posted by them. Hence users of social network need to devote a significant amount of time to maintain relationships on social networks, but frequently do not have time to do it. For close friends and family, users would still socialize manually. For the rest of the network, they would use the automated agent for social promotion being proposed.

The difficulty in solving this problem lies mainly in the area of relevance. Messages of the automated agent must be relevant to what human agents are saying. These messages are not always expected to be impressive, witty, or written in style, but at least they should show social engagement. CASP should show that its host cares about the friends being communicated with.

The opportunity to automate social promotion leverages the fact that overall coherence and exactness of social communication is rather low. Readers would tolerate worse than ideal style, discourse and quality of content being communicated, as long as overall the communication is positive and makes sense. Currently available commercial chat bots employed by customer support portals, or packaged as mobile apps, possess too limited NLP and text understanding capabilities to support conversations for social profiling.

In Fig. 12.4 we show CASP posting a message about his "experience" at lake Tahoe, having his host's friend newly posted image caption as a seed.

12.3 CASP Architecture

CASP is packaged as a chatbot: it inputs a seed (single or multiple postings) written by human peers of the host and outputs a message it forms from a content mined on the web or in another source, selected and/or adjusted to be relevant to this input posting. This relevance is based on the appropriateness in terms of content topic and also on the appropriateness in terms of mental/epistemic state: for example, it responds by an answer to a question, by a recommendation to a user host post asking for recommendations and by a question to a post mentioning that an individual would be happy to answer a question of her peer.

CASP includes the following components:

- Web mining component, which forms the web search queries from the seed and obtains search results using APIs such as Bing, Yahoo! or Yandex;
- Content relevance component, which filters out irrelevant portions of candidate content found on the web, based on syntactic generalization operator (Galitsky et al. 2011). It functions matching the parse thicket for a seed with the parse thicket for a content found on the web;
- Mental state relevance component, which extracts mental states from the seed message and from the web content and applies reasoning to verify that the former can be logically followed by the latter.

In Fig. 12.2 we show a high-level view of CASP architecture, outlining most critical components of web search for candidate postings and relevance verification.

Content relevance component is described in details in Galitsky et al. (2013) and Chap. 9. It is based on text similarity function which relies on generalization operation of syntactic, semantic and discourse-level representation of text.

In Galitsky (2013) we developed a generic software component for computing consecutive plausible mental states of human agents that is employed by CASP. The simulation approach to reasoning about mental world is based on exhaustive search through the space of available behaviors. This approach to reasoning is implemented as a logic program in a natural language multiagent mental simulator NL_MAMS, which yields the totality of possible mental states few steps in advance, given an arbitrary initial mental state of participating agents. Due to an extensive vocabulary of formally represented mental attitudes, communicative actions and accumulated library of behaviors, NL_MAMS is capable of yielding much richer set of sequences of mental state than a conventional system of reasoning about beliefs, desires and intentions would deliver (Galitsky 2016). Also, NL_MAMS functions in domain-independent manner, outperforming machine learning-based systems for accessing

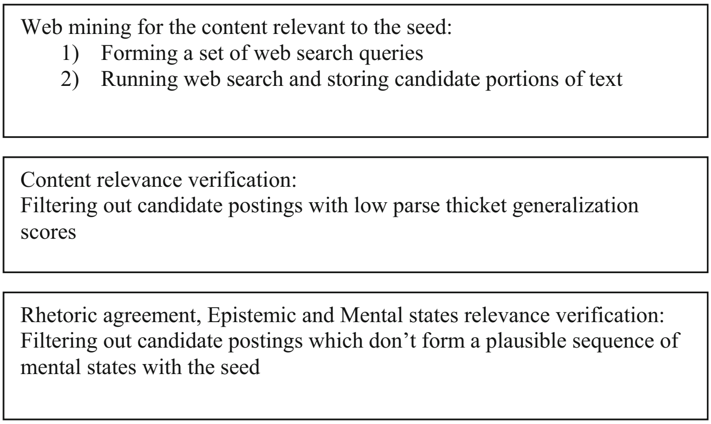

Fig. 12.2 A higher-level view of CASP components and relevance pipeline

plausibility of a sequence of mental states and behaviors of human agents in broad domains where training sets are limited (Galitsky and Shpitsberg 2016).

Detailed CASP architecture that includes all components is shown in Fig. 12.3. The leftmost column includes the posting preparation components, the column in the middle – web mining (Buzmakov 2015) and forming the list of candidate posting, and the column on the right – relevance filtering components. The bottom row of the chart includes merging textual chunks and a final delivery of the formed posting. In each row, the processing by components occurs from top to bottom.

Once CASP obtains a current state of a conversational thread, it needs to decide if / when is a good time to post. To form a query, the conversational thread should settle in terms of topic. Also, rate of postings should drop to make sure CASP does not break the thread (so that the next thread participant would need to adjust his posting).

To form a query from a single (initial, seed posting) or the whole conversational thread, CASP needs to obtain a topic, or main entity (named entity) of this single or multiple texts respectively. To do that, CASP extracts noun phrases and scores them with respect to estimated importance. For the case of multiple texts, lattice querying mechanism (Sect. 12.8) is employed to get the level of optimal generality: if it is too low, then the web mining would find too few of too specific search results which might be inappropriate. If this generality of web mining query is too high, then the resultant posting might be irrelevant, too broad, so would be hard for peers to see how CASP maintains the relevance of the conversation.

The chatbot forms multiple web mining queries since it is unclear which one would give the content from the web that would pass the relevance filtering. For each query, we form a set of search results and form a single list of candidate postings. Relevance filtering either selects the best posting or a few best ones whose selected text chunks will be combined.

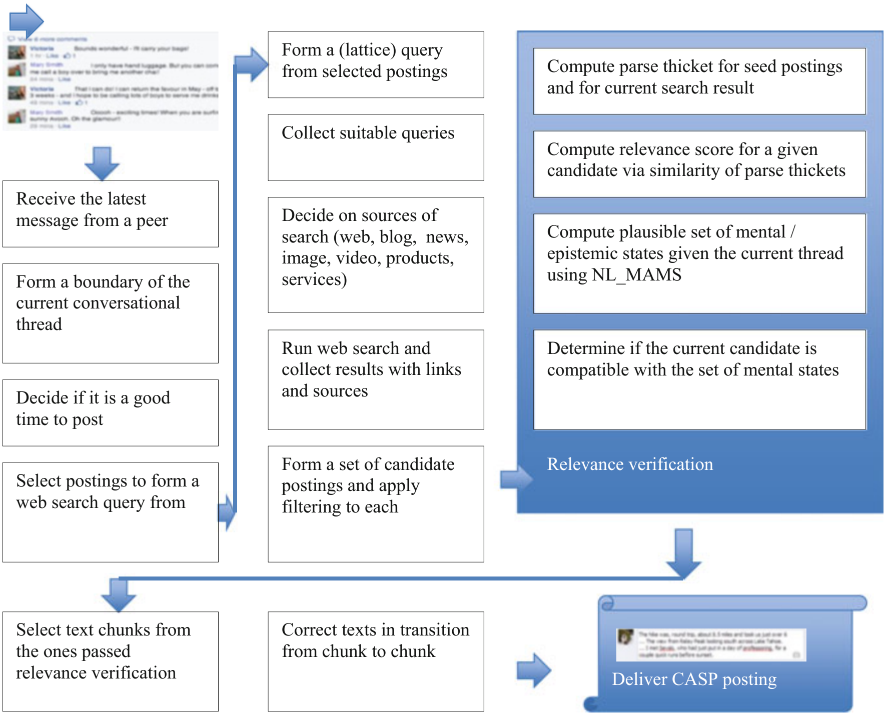

Fig. 12.3 Detailed architecture of CASP

12.4 Use Cases of CASP

We start with the use case of expressing an interest to a friend sharing his travel experience, posting a photo. CASP is looking for a content related to a place (*Lake Tahoe*) and an event (*sunset*). In this first use case, CASP finds and forms a piece of content and its human host posts it on Facebook after a couple of friends have commented. It is not disclosed that a text is formed by CASP but some irrelevant details (such as mentioning a random person) may cause suspicion (Fig. 12.4).

In the second use case, CASP greets a friend on his arrival back from her trip (Fig. 12.5). In this case CASP is explicit on representing his host so a stronger deviation of message appropriateness can be handled. CASP waits as conversation passes through a couple of cycles and then yields a message with a link covering the entire conversation, not just the first, seed posting. CASP found a relevant posting by another Facebook user (not a random web page) with an image.

The third use case (Fig. 12.6) shows how CASP can take into account mental states and sentiments of previous comments (Galitsky and McKenna 2017). Posting is somewhat relevant: it does not talk about a child unhappy with a parent singing but

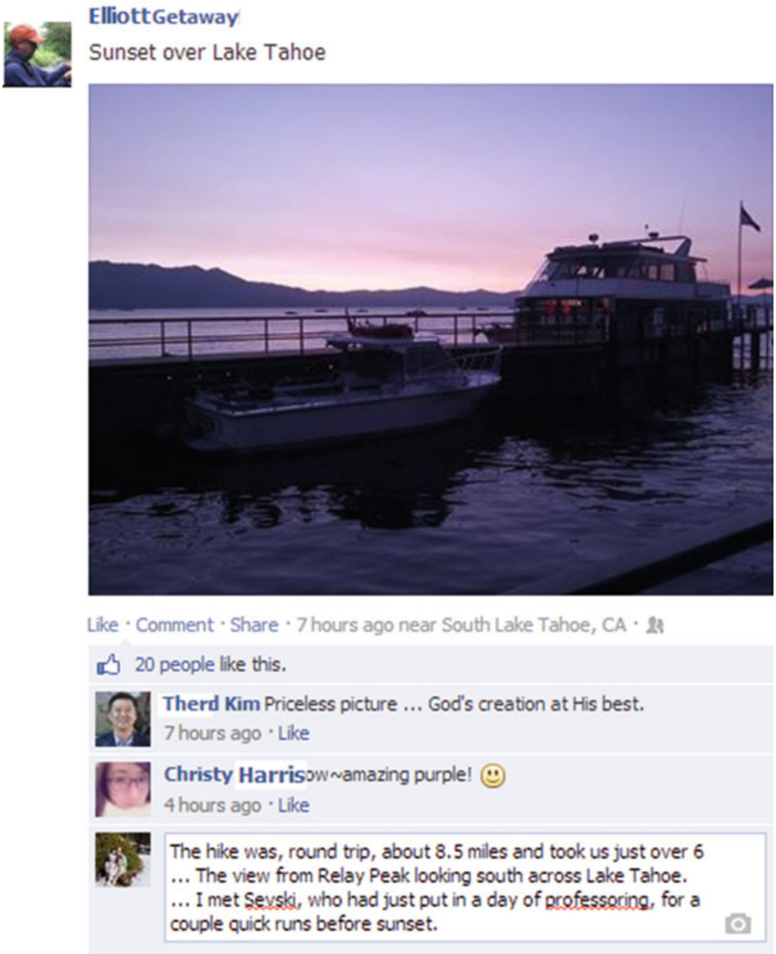

Fig. 12.4 CASP is posting a message for *Lake Tahoe sunset*

instead suggests what to sing. However, this far-reaching correlation with the seed is suitable for casual friendly conversations.

Finally, we share a case study where a posting by CASP initiated a discussion on ethics of automated postings (Fig. 12.7). Two friends posted a photo of them dancing tango. CASP commented on the posting, finding information about “tango at a wedding” (on the bottom). The friends got upset and communicated that the posting of CASP citing the wedding was irrelevant and they did not like it (on the right). The author of this book then intervened and shared his ideas on usability of CASP in response. The conversation led to the state when the parties *agreed to disagree*. Nevertheless, the couple married 5 months later.

Fig. 12.5 CASP is posting a message welcoming his friend back home, having recognized the mental state of the participants of the chat

12.5 Evaluation of Relevance

In this Section we evaluate the relevance of a CASP posting assessed by selected judges, irrespectively of how it was perceived by peers in the real-world settings (Table 12.1). We conducted evaluation of relevance of syntactic generalization–enabled search engine (Galitsky et al. 2012), based on Yahoo and Bing search engine APIs.

Fig. 12.6 CASP commenting on the posting of a friend

The value of relevance for a posting is Boolean: acceptable or not. Individual postings are assessed so no complications arise due to measuring multiple search results. We vary the complexity of seed posting and provide the percentages of relevant results found on the web and subject to relevance filtering by linguistic means. We show these percentages as the complexity of such filtering increases. Accuracy of a particular search setting (query type and search engine type) is calculated, averaging through 40 search sessions. For our evaluation, we use user postings available at author' Facebook accounts. The evaluation was done by the author. We refer the reader to (Chaps. 5 and 7) for the further details on evaluation settings for search relevance evaluation.

To compare the relevance values between search settings, we used first 30 search results and re-ranked them according to the score of the given search setting. We use three approaches to verify relevance between the seed text and candidate posting:

Fig. 12.7 A case study with Facebook friends. On the top: an original photo with the caption which was a CASP seed. On the bottom: Text and Image found by CASP. On the right: discussions between CASP's host and his friends on appropriateness of CASP posting

(a) Pair-wise parse tree matching, where the tree for each sentence from seed is matched with the tree for each sentence in the candidate posting mined on the web;
(b) The whole graph (parse thicket) for the former is matched against a parse thicket for the latter using phrase-based approach. In this case parse thickets are represented by all their paths (thicket phrases, Chap. 7);
(c) The match is the same as (2) but instead of phrases we find a maximal common subgraph (Chap. 5).

Table 12.1 Evaluation results for various search domains and for various implementations of PT generalization

Query complexity	Relevance of baseline Bing search, %, averaging over 40 searches	Relevance of PT/phrase generalization search, %, averaging over 40 searches, using original text, without SpAtcT	Relevance of PT/phrase generalization search, %, averaging over 40 searches, using snippets	Relevance of PT/phrase generalization search, %, averaging over 40 searches, using original text	Relevance of PT/graph generalization search, %, averaging over 40 searches, using snippets	Relevance of PT/graph generalization search, %, averaging over 40 searches, using original text
1 compound sent	54.5	61.3	63.3	65.3	66.2	67.2
2 sent	52.3	60.9	60.7	62.1	63.4	63.9
3 sent	49.7	55.4	61.7	61.9	60.8	61.9
4 sent	50.9	55.5	60.5	61.1	61.5	62.7
Average	51.85	58.28	61.55	62.6	62.98	63.93

The value of parse thicket based generalization (Chap. 7) varies from domain to domain significantly, mostly depending on the writing style and use of terminology by the authors presenting their opinions on the products. When things in a domain are named uniquely, and the typical writing style is plain enumeration of product features, contribution of parse thickets is the least (shopping product domains). On the contrary, where writing styles vary a lot and different people name the same things differently, in such horizontal domain as Facebook, the baseline relevance is low, the resultant relevance is lower (63%) than in the other domains (73–75%) but matching of parse thickets helps in a higher degree.

Proceeding from snippets to original paragraph(s) in a webpage gives further 0.8% increase for both thicket phrase-based and graph-based computation of PT.

One can observe that unfiltered precision is 52%, whereas improvement by pair-wise sentence generalization is 11%, thicket phrases – additional 6%, and graphs – additional 0.5%. Hence the higher the complexity of sentence, the higher is the contribution of generalization technology, from sentence level to thicket phrases to graphs.

12.6 Evaluation of Extraction of Communicative Action

To learn similar sequences of communicative actions from text, we need to be capable of extracting them. We conduct the evaluation for the complex information extraction task such as identifying communicative actions and detecting emotional states (Galitsky and Tumarkina 2004). Also, we perform evaluation for the rhetoric relation domain: this task is necessary to build a set of parse trees for a paragraph, linking its parse trees into PT. This approach is pre- communicative discourse trees (CDT) that was introduced in Chap. 10. We rely on the following information extraction techniques:

- Keyword- and regular expression – based string match;
- Keyword- and regular expression – based Boolean Lucene queries;
- Lucene Span queries where the distance between keywords in text is constrained;
- Lattice query-based information extraction, where the template is automatically generalized from multiple parse trees for occurrences of a given communicative action.

The corpus is based on the set of customer complains (Chap. 13), where both communicative actions and emotions are frequent and essential for complaint analysis tasks. Evaluation was conducted by quality assurance personnel. The first two information extraction settings are baseline, the third can be considered as an industry standard, and the last one is designed to be a state-of-the-art for extracting communicative actions in their explicit form such as communicating verbs as well as various implicit forms.

We observe in Table 12.2 that the information extraction F-measure for Keywords and Regular expressions is both 64% for querying indexed data and string search, averaging through our extraction domains. Relying on span and ‘like’ queries gives

Table 12.2 Evaluation of communicative action extraction task

Method task	Keywords and Regexps via string match		Keywords and Regexp queries		Span and 'like' queries		PT-based extraction rules	
	P/R		P/R		P/R		P/R	
Communicative actions	64	71	63	72	68	70	82	75
Mental and emotional states	62	70	59	70	64	68	80	74

just 2% increase in F-measure, whereas using frame queries delivers further 10% improvement. Communicative actions give just 2–3% better performance than mental states, and rhetoric structures improve the accuracy by further 3–5%.

12.7 Evaluation of Trust

Primarily, the host human agent should trust the social promotion agent CASP that the results of its activity would indeed improve the host's position in social space, not decrease it. Relying on an incompetent, silly CASP may lead to unwanted consequences such as a drop in the reputation of the CASP host (Galitsky and Levene 2007). The promotion agent targets least important friends and members of the network, however if a significant portion of them lose trust in the host agent, the overall impact of the social promotion campaign would become negative. If a human host loses the trust in her auto promotional agent, she would stop using it.

Secondarily, the friends and members of social network may lose trust in the host agent irrespectively of how the communication has been facilitated, and may unfriend the host agent or block his messages. This might happen because of a loss of relevance, loss of rhetorical appropriateness of messages and also because they can be offended by the content being communicated. From the standpoint of CASP it is most likely a problem of relevance, however the perception of irrelevant messages can be ambiguous. Friends can think of such message as a bad joke, a hint for something they would not want to share, and even as an insult.

There are two following cases the friends and members of the social network of a host loose trust in the host agent himself when he is using CASP:

- If they do not know that an agent acts on his behalf, they may get upset by irrelevance and inappropriateness of communication without making the reason for it clear. They would consider it insulting to use such communication means as CASP instead of direct human-human communication;
- If they know that they receive message from an automated agent, but the results are less relevant and less appropriate than what they expected. We have encountered this case in Fig. 12.7.

We now share our data on how some peers have been loosing trust in as much degree as stopping using CASP at all and even unfriending its human host. We do

Table 12.3 The data on the number of irrelevant postings till an occurrence of certain dissatisfaction event

Topic of the seed	Complexity of the seed and posted message	A friend complains to the CASP's host	A friend unfriends the CASP host	A friend shares with other friends that the trust in CASP is lost in one way or another	A friend encourages other friends to unfriend a friend with CASP
Travel & outdoor	1 sent	6.3	8.5	9.4	12.8
	2 sent	6.0	8.9	9.9	11.4
	3 sent	5.9	7.4	10.0	10.8
	4 sent	5.2	6.8	9.4	10.8
Shopping	1 sent	7.2	8.4	9.9	13.1
	2 sent	6.8	8.7	9.4	12.4
	3 sent	6.0	8.4	10.2	11.6
	4 sent	5.5	7.8	9.1	11.9
Events & entertainment	1 sent	7.3	9.5	10.3	13.8
	2 sent	8.1	10.2	10.0	13.9
	3 sent	8.4	9.8	10.8	13.7
	4 sent	8.7	10.0	11.0	13.8
Job-related	1sent	3.6	4.2	6.1	6.0
	2 sent	3.5	3.9	5.8	6.2
	3 sent	3.7	4.0	6.0	6.4
	4 sent	3.2	3.9	5.8	6.2
Personal life	1 sent	7.1	7.9	8.4	9.0
	2 sent	6.9	7.4	9.0	9.5
	3 sent	5.3	7.6	9.4	9.3
	4 sent	5.9	6.7	7.5	8.9
Average		6.03	7.50	8.87	10.58

not see a reason of stopping using CASP other than loosing trust and starting perceiving the CASP-facilitated conversation as unfaithful, loosing an intimacy of friendship, abusing privacy and so forth. To track how the peer users loose trust as they encounter more CASP activity, we firstly report the *number* of such encounters associated with negative user experience till the user reaches the respective level of mistrust (Table 12.3). After that, we measure the level of relevance that leads to this level of mistrust. Whereas the first dataset does not measure irrelevance and instead reports the number of irrelevant scenarios, the second dataset does the other way around and provides an explicit relevance data.

After a certain number of CASP failures to provide relevant postings, friends *loose trust and complain, unfriend, shares negative information* about the lost of trust with others and even *encourage other friends to unfriend* a friend who is enabled with CASP (Table 12.3). The values in the cells indicate the average number of postings with failed relevance when the respective event of disengagement from CASP occurred. These posting of failed relevance were tracked within 1 months of the experiment run, and we do not access the values for the relative frequency of occurrences of these postings. On average, 100 postings were done for each user (1–4 CASP postings per a seed posting).

One can see that in various domains the scenarios of users' tolerance to irrelevance varies. For less information-critical domains like *travel* and *shopping*, this tolerance to failed relevance is relatively low. Conversely, in the domains taken more seriously by peers, like *job* related, and the domains with personal flavor and increased privacy, like *personal life*, users are more sensitive to CASP failures and the lost of trust in its various forms occur faster. For all domains, tolerance slowly decreases when the complexity of posting increases. Users' perception is worse for longer texts, irrelevant in terms of content or their expectations, than for shorter, single sentence or phrase postings by CASP.

We now drill into the types of relevance errors which lead to deterioration of trust by peer users of CASP. We outline the following cases where a CASP posting is rejected by recipients:

(a) The content CASP is posted is topically irrelevant to the content of original post by a human friend;
(b) CASP content is topically relevant to the content, but irrelevant in terms of style, user knowledge (epistemic states), user beliefs (in such domain as politics). This form of relevance is referred to as rhetorical agreement and explored in Chap. 10.

In Table 12.4 we focus on the user tolerance vs irrelevance data in the same format as above (Table 12.3) but measuring relevance values, for both (a) and (b). We use a Boolean value for relevance: either relevant or totally irrelevant posting. For each level of dissatisfaction, from complaint to encouraging others, we measure the value of relevance where at least 20% of the peers reach this level, given the domain and complexity and/or size of CASP posting. For example, in *travel* domain, for 1 sentence posting, more than 20% of the peers start to complain to the CASP host when relevance goes as lows as 83% (17 percent of postings are irrelevant).

One can see from Table 12.4 that the users can tolerate stronger problems with rhetorical agreement and epistemic states than with content relevance. As the complexity and /or length of posting grows, users can tolerate lower relevance. There is a few percent (3–10) drop of either content relevance or communicative actions plausibility where a user dissatisfaction becomes more severe; it depends on the problem domain. For job-related communications, user sensitivity to problems with both kinds of relevance is higher than for *travel, entertainment* and *personal life* domains (Fig. 12.8).

Now we compare indirect relevance assessment in Table 12.1 and failed relevance assessment in this section (Table 12.4). Out of hundred CASP posting per user who made between 2 and 3 manual postings, failures occurred in less then 10% of CASP postings. Therefore most peer users do not end up refusing CASP posting, having their trust of it lost. The friends who were lost due to the abuse of their tolerance to meaningless postings by CASP would become inactive CASP users in most cases anyway (because of a lack of attention and interest to the CASP host).

Table 12.4 The data on the percentage of irrelevant postings till an occurrence of certain dissatisfaction event

Topic of the seed and posting / degrees of user tolerance	Complexity of the seed and posted message	A friend complaints to the CASP's host	A friend unfriends the CASP host	A friend shares with other friends that the trust in CASP is lost	A friend encourages other friends to unfriend a friend with CASP
Travel & outdoor	1 sent	83/67	76/63	68/60	61/53
	2 sent	81/68	74/62	75/59	59/54
	3 sent	78/66	74/64	64/58	57/50
	4 sent	75/63	70/62	60/59	55/50
Events & entertainment	1 sent	86/70	79/67	74/65	71/60
	2 sent	82/70	78/66	72/61	69/58
	3 sent	79/69	76/67	74/64	67/59
	4 sent	78/68	76/66	73/63	65/60
Job-related	1sent	80/67	77/63	66/55	59/51
	2 sent	77/65	73/61	70/54	56/51
	3 sent	75/63	71/65	63/56	55/48
	4 sent	74/60	68/63	61/57	56/51
Personal life	1 sent	82/66	75/64	66/62	57/50
	2 sent	80/66	73/65	70/57	60/52
	3 sent	78/62	75/62	66/56	58/48
	4 sent	77/60	75/58	68/55	59/52

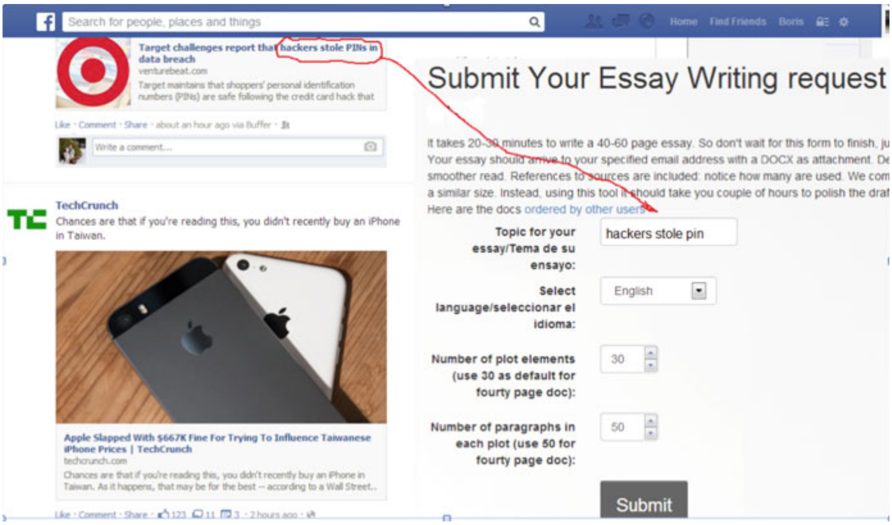

Fig. 12.8 A front-end for the 'on-demand' reply generation; Facebook prompt is on the left. The form to specify the format, size and language of the desired content is on the right

However, luckily, a majority of social network friends will be retained and stay in an active mode, keeping receiving CASP postings.

12.8 Replying to Multiple Posts

When a single seed text is used to generate a query, we just identify its noun phrases and named entities and form a web mining query from them. When CASP chatbot relies on multiple texts from a conversational thread, we need to selects phrases and entities that represent the topic of the whole conversation, not just the topic of an initial posting. To obtain an expression for this topic, we need to control the level of generality, attempting to generalize these multiple texts, and a new technique referred to Lattice Querying is coming into play.

12.8.1 Introducing Lattice Querying

Today, it is hard to overestimate the popularity of information access via search engines. Also, a vast number of distributed computing frameworks have been proposed for big data. They provide scalable storage and efficient retrieval, capable of collecting data from various sources, fast moving and fairly diverse. Modern open source big data search and exploration systems like SOLR and ElasticSearch are broadly used for access and analysis of big data. However, intelligence features such as search relevance and adequate analysis, retrieval and exploration of large quantities of natural language texts are still lacking. Therefore for a social promotion chatbot it is still hard to rely on available search engines to yield a high volume of meaningful posts. Modern search engines and libraries still treat a query as a bag of words with their statistics. In spite of the extensive capabilities of natural language parsing, they are still not leveraged by most search engines.

Also, frequently novice users of search engines experience difficulties formulating their queries, especially when these queries are long. It is often hard for user who is new to a domain to pick proper keywords. Even for advanced users exploring data via querying, including web queries, it is usually hard to estimate proper generality / specificity of a query being formulated. Lattice querying makes it easier for a broad range of user and data exploration tasks to formulate the query: given a few examples, it formulates the query automatically.

In this Section we introduce a proactive querying mode, when a chatbot finds information for its human host automatically. We intend to leverage the efficiency of distributed computing framework with the intelligence features of data exploration provided by NLP technologies. We introduce the technique of lattice querying which automatically forms the query from the set of text samples provided by a user by generalizing them from the respective parse trees. Also, the system produces search results by matching parse trees of this query with that of candidate answers. Lattice

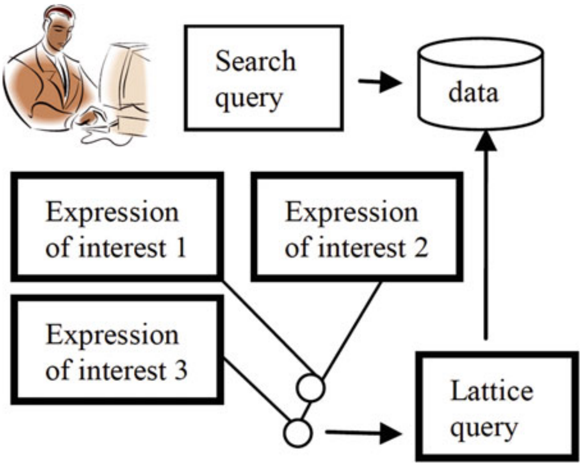

Fig. 12.9 A lattice query in comparison with a regular query

queries allow increase in big data exploration efficiency since they form multiple *hypotheses* concerning user intent and explore data from multiple angles (generalizations).

Exploring data, mostly keyword query and phrase query are popular, as well as natural language-like ones. Users of search engines also appreciate 'fuzzy match' queries, which help to explore new areas where the knowledge of exact keywords is lacking. Using synonyms, taxonomies, ontologies and query expansions helps to substitute user keywords with the domain-specific ones to find what the system believes users are looking for Ourioupina and Galitsky (2001) and Galitsky (2003). Lattice queries increase usability of search, proceeding from expressions in user terms towards queries against available data sources.

The idea of lattice query is illustrated in Fig. 12.9. Instead of a user formulating a query exploring a dataset, he provides a few samples (expressions of interest) so that the system formulates a query as an overlap (generalization) of these samples, applied in the form of a lattice (shown in bold on the bottom).

Proceeding from a keyword query to regular expressions or fuzzy one allows making search more general, flexible, assists in exploration of a new domain, as set of document with unknown vocabulary. What can be a further step in this direction? We introduce lattice queries, based on natural language expressions that are generalized (Chap. 5) into an actual query.

Nowadays, search engines ranging from open source to enterprise offer a broad range of queries with string character-based similarity. They include Boolean queries, span queries which restrict the distances between keywords in a document, regular expressions queries which allow a range of characters at certain positions, fuzzy match queries and more-like-this which allow substitution of certain characters based on string distances. Other kinds of queries allow expressing constraints in a particular dimension, such as geo-shape query. Proceeding from a keyword query to regular expression or fuzzy one allows making search more general, flexible, assists in exploration of a new domain, such as a set of document with unknown vocabulary. What can be a further step in this direction? We introduce lattice queries, based on natural language expressions that are generalized into an actual query. Instead of getting search results similar to a given expression (done by 'more like this' query), we first build the commonality expression between all or subsets of the given sample expressions, and then use it as a query. A lattice query includes words as well as attributes such as entity types and verb attributes.

Forming lattice queries is based on Generalization operation introduced in Chap. 5.

12.8.2 Sentence-Based Lattice Queries

Let us start with an employee search example; imagine a company looking for the following individuals:

- *A junior sale engineer expert travels to customers on site;*
- *A junior design expert goes to customer companies;*
- *A junior software engineer rushes to customer sites.*

Given the above set of samples, we need to form a job-search query that would give us candidates somewhat similar to what we are looking for. A trivial approach would be to just turn each sample into a query and attempt to find an exact match. However most of times it would not work, so such queries need to release some constraints. How to determine which constraints need to be dropped and which keywords are most important?

To do that, we apply generalization to the set of these samples. For the entities and attributes, we form the least general generalization. The seniority of the job (adjective) '*junior*' will stay. The job activity (noun phrase) varies, so we generalize them into <*job-activity*>. The higher-level reference to the job is '*expert*' and is common for all three cases, so it stays. The verbs for *job responsibility* vary, so we use <*action*> that can be further specified as

<*moving_action*>, using verb-focused ontologies like VerbNet. To generalize the last noun phrase, we obtain the generalization <*customer, NP*>:
junior <*any job activity*> *expert* <*action*> *customer-NP.*

This is a lattice query, which is expected to be run against *job descriptions* index and find the cases which are supposed to be most desired, according to the set of samples.

In terms of parse trees of the potential sentences to be matched with the lattice query, we rewrite it as

JJ-junior NP- NN-expert VP-* NN-customer NP-**

The lattice query read as *find me a junior something expert doing-something-with customer of-something*.

```
[[<1>NP'A':DT, <2>NP'junior':JJ, <3>NP'sale':NN, <4>NP'engineer':NN,
<5>NP'expert':NN], [<6>VP'travels':VBZ, <7>VP'to':TO, <8>VP'customers':NNS,
<9>VP'on':IN, <10>VP'site':NN], [<7>PP'to':TO, <8>PP'customers':NNS, <9>PP'on':IN,
<10>PP'site':NN], [<8>NP'customers':NNS, <9>NP'on':IN, <10>NP'site':NN],
[<8>NP'customers':NNS], [<9>PP'on':IN, <10>PP'site':NN], [<10>NP'site':NN]]

[[<1>NP'A':DT, <2>NP'junior':JJ, <3>NP'design':NN, <4>NP'expert':NN],
[<5>VP'goes':VBZ, <6>VP'to':TO, <7>VP'customer':NN, <8>VP'companies':NNS],
[<6>PP'to':TO, <7>PP'customer':NN, <8>PP'companies':NNS], [<7>NP'customer':NN,
<8>NP'companies':NNS]]

[[<1>NP'A':DT, <2>NP'junior':JJ, <3>NP'software':NN, <4>NP'engineer':NN],
[<5>VP'rushes':VBZ, <6>VP'to':TO, <7>VP'customer':NN, <8>VP'sites':NNS],
[<6>PP'to':TO, <7>PP'customer':NN, <8>PP'sites':NNS], [<7>NP'customer':NN,
<8>NP'sites':NNS]]
```

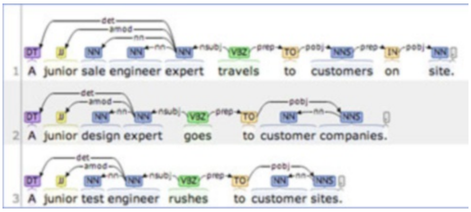

Fig. 12.10 Parse trees and phrase representation for three samples to form a lattice query

Now we show how this template can be applied to accept/reject a candidate answer *Cisco junior sale representative expert flew to customers data centers*.

We represent the lattice query as a conjunction of noun phrases (NP) and verb phrases (VP) set:

[[NP [DT-a JJ-junior NN- NN-*], NP [NN*-customers]], [VP [VB-* TO-to NN*-customers]]]*

The first NP covers the beginning of the lattice query above, and the second NP covers the end. VP covers the second half of the lattice query starting from *doing-something...*

The generalization between the lattice query and a candidate answer is

[[NP [JJ-junior NN- NN-*], NP [NN*-customers]], [VP [VB-* TO-to NN*-customers]]]*

One can see that the NP part is partially satisfied (the article a does not occur in the candidate answer) and VP part is fully satisfied.

Here are the parse trees for three samples (Fig. 12.10):

Generalizing these three expressions, we obtain the lattice query to run against a dataset:

[[NP [DT-a JJ-junior NN- NN-*], NP [NN*-customers]], [VP [VB-* TO-to NN*-customers]]]*

One can see that using lattice queries, one can be very sensitive in selecting search results. Searching for a token followed by a word with certain POS instead of just a single token gives a control over false-positive rate. Automated derivation of such

constraint allows a user to focus on specific cases instead of making efforts to generate a query which would keep the expected search results in and unwanted out.
Definition: a lattice query Q is satisfied by a sentence S, if Q^$S = S$.
In practice a weak satisfaction is acceptable, where

Q^$S \in S$, but there are constraints on the parts of the lattice query:

- A number of parts in Q^S should be the same as in Q;
- All words (not POS-* placeholders) from Q should also be in Q^S.

12.8.3 Paragraph-Level Lattice Queries

Text samples to form a lattice query can be typed, but also can be taken from an existing text. To expand the dimensionality of content exploration, samples can be paragraph-size texts (Galitsky 2014).

Let us consider an example of a safety-related exploration task, where a researcher attempts to find a potential reason for an accident. Let us have the following texts as incidents descriptions. These descriptions should be generalized into a lattice query to be run against a corpus of texts for the purpose of finding a root cause of a situation being described.

Crossing the snow slope was dangerous. They informed in the blog that an ice axe should be used. However, I am reporting that crossing the snow field in the late afternoon I had to use crampons.

I could not cross the snow creek since it was dangerous. This was because the previous hiker reported that ice axe should be used in late afternoon. To inform the fellow hikers, I had to use crampons going across the show field in the late afternoon.

As a result of generalization from two above cases, we will obtain a set of expressions for various ways of formulating commonalities between these cases. We will use the following snapshot of a corpus of text to illustrate how a lattice query is matched with a paragraph:

I had to use crampons to cross snow slopes without an ice axe in late afternoon this spring. However in summer I do not feel it was dangerous crossing the snow.

We link two phrases in different sentences since they are connected by a rhetoric relation based on *However* . . .

```
rel: <sent=1-word=1..inform> ===> <sent=2-word=4..report>
From [<1>NP'They':PRP]
TO [<4>NP'am':VBP, NP'reporting':VBG, <8>NP'the':DT,
<9>NP'snow':NN, <10>NP'field':NN, <11>NP'in':IN, <12>NP'the':DT,
<13>NP'late':JJ, <14>NP'afternoon':NN, <15>NP'I':PRP,
<16>NP'had':VBD, <17>NP'to':TO, <18>NP'use':VB,
<19>NP'crampons':NNS]
```

We are also linking phrases of different sentences based on communicative actions:

```
rel: <sent=1-word=6..report> ===> <sent=2-word=1..inform>
From [<4>NP'the':DT, <5>NP'previous':JJ, <6>NP'hiker':NN]
TO [<1>NP'To':TO, <2>NP'inform':VB, <3>NP'the':DT,
<4>NP'fellow':JJ, <5>NP'hikers':NNS]
```

As a result of generalizing two paragraphs, we obtain the lattice query:

```
[[NP [NN-ice NN-axe ], NP [DT-the NN-snow NN-* ], NP [PRP-i ], NP
[NNS-crampons ], NP [DT-the TO-to VB-* ], NP [VB-* DT-the NN-*
NN-field IN-in DT-the JJ-late NN-afternoon (TIME) ]], [VP [VB-was
JJ-dangerous ], VP [VB-* IN-* DT-the NN-* VB-* ], VP [VB-* IN-*
DT-the IN-that NN-ice NN-axe MD-should VB-be VB-used ], VP [VB-*
NN-* VB-use ], VP [DT-the IN-in ], VP [VB-reporting IN-in JJ-late
NN-afternoon (TIME) ], VP [VB-* NN*-* NN-* NN*-* ], VP [VB-crossing
DT-the NN-snow NN-* IN-* ], VP [DT-the NN-* NN-field IN-in DT-the
JJ-late NN-afternoon (TIME) ], VP [VB-had TO-to VB-use
NNS-crampons ]]]
```

Notice that potential safety-related ‘issues’ are *ice-axe, snow, crampons, being at a . . . field during later afternoon, being dangerous, necessity to use ice-axe, crossing the snow*, and others. These issues occur in both samples, so they are of a potential interest. Now we can run the formed lattice query against the corpus and observe which issues extracted above are confirmed. A simple way to look at it is as a Boolean OR query: find me the conditions from the list which are satisfied by the corpus. The generalization for the lattice query and the paragraph above turns out to be satisfactory:

```
[[NP [NN-ice NN-axe ], NP [NN-snow NN*-* ], NP [DT-the NN-snow ], NP
[PRP-i ], NP [NNS-crampons ], NP [NN-* NN-* IN-in JJ-late
NN-afternoon (TIME) ]], [VP [VB-was JJ-dangerous ], VP [VB-* VB-use
], VP [VB-* NN*-* IN-* ], VP [VB-crossing NN-snow NN*-* IN-* ], VP
[VB-crossing DT-the NN-snow ], VP [VB-had TO-to VB-use
NNS-crampons ], VP [TO-to VB-* NN*-* ]]] => matched
```

Hence we got the confirmation from the corpus that the above hypotheses, encoded into this lattice query, are true. Notice that forming a data exploration queries from the original paragraphs would contain too many keywords and would produce too much marginally relevant results.

12.8.4 Evaluation of Web Mining via Lattice Queries

We evaluate the data exploration scenarios using search engine APIs. Instead of formulating a single complex question and submit it for search, a user is required to describe her situation in steps, so that the system would assist with formulating hypotheses on what is important and what is not. The system automatically derives generalizations and builds the respective set of lattice queries. Then the search engine API is used to search the web with lattice queries and automatically filter out results which are not covered by the lattice query. To do the latter, the system generalizes each candidate search results with each lattice query element and rejects the ones not covered, similar to the information extraction scenario.

This year I purchased my Anthem Blue Cross insurance through my employer. What is the maximum out-of-pocket expense for a family of two in case of emergency?
Last year I acquired my individual Kaiser health insurance for emergency cases only. How much would be my out of pocket expense within a year for emergency services for my wife and kids?

The system finds a commonality between these paragraphs and forms a lattice query, so that the search results are as close to this query as possible. An alternative approach is to derive a set of lattice queries, varying generalization results, and delivering those search results which are covered the best with one of the lattice query from this set (not evaluated here). A Bing search results for the query '*out-of-pocket expense health insurance emergency*' is shown in Fig. 12.11 (API delivers the same results).

We show the percentage of relevant search results, depending on how queries are formed, in Table 12.5. We ran 20 queries for each evaluation setting and considered first 20 results for each. Each search results is considered as either relevant or not, and we do not differentiate between top search results and 15th–20th ones. We use Bing search engine API for these experiments. Evaluation of lattice querying on the web was conducted by the author.

One can see that for the sentence-level analysis, there is 14% improvement proceeding from keyword overlap to parse structures delivering phrases for web search, and further 8% improvement leveraging lattice queries derived from a pair of sentences. For the paragraphs, there are respective 21% and 22% improvement, since web search engines do not do well with paragraph-sized queries. If the number of keywords in a query is high, it is hard for a search engine to select which keywords are important, and term frequency becomes the major ranking factor. Also, for such queries, a search engine cannot rely on learned user selections from previous querying, hence the quality of search results are so low.

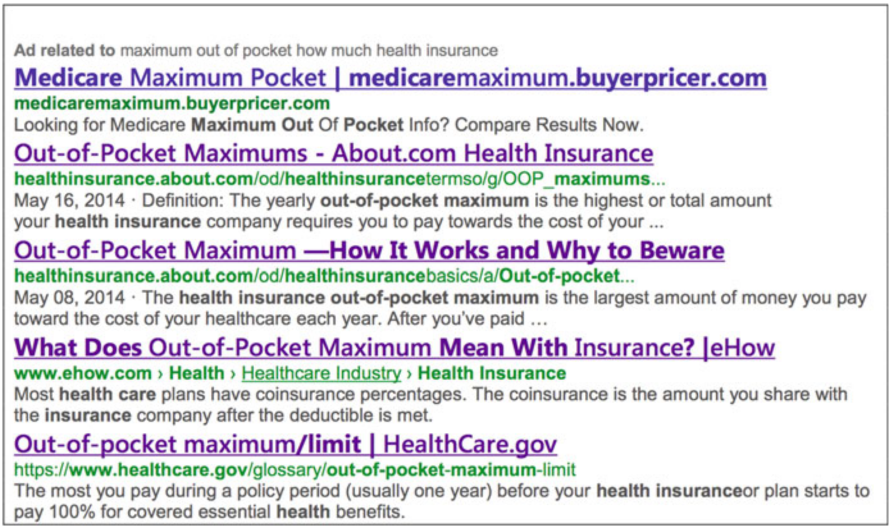

Fig. 12.11 Once a lattice query is formed from samples, we obtain search results from the web using search API

Table 12.5 Evaluation of web mining

Method task	Forming lattice query as keyword overlap for two sentences	Forming lattice query as parse structure of a sentence	Lattice queries for two sentences	Forming lattice query as keyword overlap for paragraphs	Forming lattice query as parse structure	Lattice queries for two paragraphs
Legal research	59	62	70	43	51	62
Marketing research	55	68	69	46	53	64
Health research	52	65	71	42	55	67
Technology research	57	63	68	45	53	64
History research	60	65	72	42	52	65

The proposed technique seems to be an adequate solution for cross-sentence alignment (Chambers et al. 2007; MacCartney et al. 2008). One application of this problem is automated solving of numerical equations formulated as algebra word problems (Kushman et al. 2014). To form a representation for an elementary algebra problem *text*, we would use a training set of pairs *textT* – *equationT* and produce an alignment of *text* and *textT* by means of generalization *text* ^ *text* (Chap. 5) that is an expression to be converted into a numerical expression. The capability to "solve" an algebraic problem is based on the completeness of a training set: for each type of equation, there should be a textual algebraic problem for it. Also, the problem of phrase alignment for such areas as machine translation has been explored in Jiang and Conrath (1997).

Let us consider an algebra problem

An amusement park sells adult tickets for $3 and kids tickets for $2, and got the revenue $500 yesterday.

We attempt to find a problem from our training set, such as:

A certified trainer conducts training for adult customers for $30 per hour and kid customer for $20 per hour, and got the revenue $1000 today.

Generalization looks like the following, including the placeholders for the values

[[NP [JJ-adult NNS- IN-for $-$ CD-3 CC-and NN*-kids NN*-*], NP [IN-* NN*-*], NP [DT-the NN-revenue $-$ CD-*]],
[VP [NN*-* IN-for $-$ CD-3 ,-, CC-and VB-got DT-the NN-revenue $-$ CD-* NN-* (DATE)], VP [CC-and NN*-kids NN*-* IN-for $-$ CD-2 CC-and VB-got DT-the NN-revenue $-$ CD-* NN-* (DATE)], VP [NN*-* IN-for $-$ CD-3 CC-and NN*-kids NN*-* IN-for $-$ CD-2]]].*

The space of possible equations can be defined by a set of equation templates, induced from training examples. Each equation template has a set of placeholders, CD-placeholders are matched with numbers from the text, and unknown placeholders are matched with nouns. Kushman et al. (2014) define a joint log-linear distribution over the system of equations with certain completeness properties. The authors learned from varied supervision, including question answers and equation systems, obtained from annotators. Features used are unigrams and bigrams, question object/sentence, word lemma nearby constant, what dependency path contains (word or another dependency path), and others, as well as equation features.

On the contrary, we rely on linguistic discourse (parse trees and their connecting arcs) to find the matching element in the training set. It is expected to shed the light on the linguistic properties of how a sentence can be converted into a part of an algebra equation.

Borgida and McGuinness (1996) proposed a declarative approach that extends standard interface functionality by supporting selective viewing of components of complex objects. Instead of just returning sets of individual objects, the queries match concepts and altered fragments of descriptions. The query language is an extended form of the language used to describe the knowledge base contents, thus facilitating user training. The term 'Frame Querying' has been used in knowledge representation framework: frame-based knowledge representation and reasoning systems typically provide procedural interfaces for asking about properties of individual objects and concepts.

12.9 Correction of Obtained Post Candidate

We will start our consideration for how to use the wisdom of the web to correct CASP postings with the inappropriate phrasings that are odd or hard to interpret. We focus on the case of machine translation that frequently gives noisy, meaningless results that cannot be posted as they are. Let us look at an example of a translation from English to Russian http://translate.google.com/#en/ru/I%20liked%20swimming%20with%20you%20a%20lot

'I liked swimming with you a lot' → 'Мне понравилось плавать с вас много' [meaning: I liked to do a high quantity of swimming with you].

This translation is not good; it demonstrates a *word→word* approach, employed by machine translation, that ignores the context of 'a lot' such as '*swim*'. This translation example works poorly with any verb, not just '*swim*'. Moreover, one can see from this example that the machine translator does not attempt to find similar Russian phrases to make sure translation results are plausible. This example is very simple, which means there should be someone on the web somewhere who said something similar. Since a machine translation itself does not usually attempt to improve the quality by verifying a translation result via web mining, we will enable CASP with this feature.

We extract phrases from "Мне понравилось плавать с вас много" and observe which phrases are found (and therefore can be confirmed) and which phrases are not found or rarely found (which means they are suspicious and perhaps need to be substituted by the ones from the web). Here is the example of web mining for phrases: https://www.google.ru/search?q="плавать+с+вас+много". All results have *плавать* in one sentence, and *с+вас* in another sentence, which means that this phrase is implausible. Now compare with https://www.google.ru/search?q="плавать+с+вами" that confirms the plausible phrase. So in the case above at least we correct the translation result into *Мне понравилось плавать с* ***вами.***

Why do we need parse thickets for text correction via web mining? Once we have more complex web mining cases, where for a few sentences we search for longer, multi-phrase search results, we need to match multiple sentences, not just phrases. For that we need some rules for how phrases can be distributed through multiple sentences. Since certain phrases can migrate from one sentence to another, we need discourse, parse thicket - level considerations to assess which modifications of sentences are plausible and which are implausible, for the purpose of match.

We designed parse thickets so that we can treat paragraph of text formally for a broad range of applications, from search to content generation. When we match two sentences, we need the rules of phrases transformation into a similar form: it is well explored and used area. When we match two paragraph of text, we need sentence parts transformation rules, which are based on RST, Speech Acts and other discourse theories we can potentially use in the future.

In machine translation, a parse thicket matching via web mining would help to assess how coherent the translation results are, based on our hypothesis that "*everything on Earth has already been said*". We match the translation results paragraph with the one found as a search result. Translation system frequently obtains meaningless output for a reasonable input. We attempt to repair such translation by trying to verify meaningfulness of each phrase in translation results. Once a meaningless phrase is detected, we form a query from its most informative keywords (Chap. 14) and search the web for most similar phrases. We assume that a majority of highly ranked texts in the web search engine results is meaningful. Once we obtain web search results for a meaningless phrase, we attempt to substitute entities' attributes (such as '*liked a lot*') or even entities themselves with the ones from the meaningless phrases to be replaced. Parse Thickets are fairly helpful in this operation, supporting the insertion of mined phrases (assumed to be meaningful) into the translation results. Finally, we expect to obtain overall more meaningful translation of a CASP posting, but potentially with a higher deviation from the original.

12.9.1 Meaningless Phrases Substitution Algorithm

We outline the CASP posting correction algorithm in the chart Fig. 12.12. For each component we provide a background and motivations for our choice of steps.

We start with forming phrases to be verified. The list of phrases that contain at least two sub-phrases is formed (simpler phrases are too trivial to verify for meaningfulness). If a phrase is too short, it will almost always be found on the web. If a phrase is too long, then even for a meaningful expression it would be hard to find similar expressions on the web, in most cases. As a result of this step, we form a list of overlapping phrases L_{op} some of which will need to be replaced. We iterate through the members of L_{op}. For a pair of consecutive overlapping phrases in L_{op}, if the first one is found to be meaningless and is replaced, the second one will not be attempted with respect to replacement.

If two consecutive phrases are still too short (< 5 words each) we merge them and insert into $\mathrm{L_{op}}$. From now on we call the elements of L_{op} expressions since they are not necessarily noun, verb or other kind of phrases.

Once the expressions are formed, we search for each expression on the web (using, for example, Bing search engine API). We first do an exact search, wrapping the expressions in double quotes. If there are no search results, we search the expression as a default (OR) query and collect the search results.

To determine if there is a similar phrase on the web or not, we assess the similarity between the expression from L_{op} and its search results. To do that, we perform generalization between the expression and each of its search result, and obtain its score (Chap. 5). For each search result, we use the highest generalization score for:

- Document title;
- Document snippet;
- Document sentence.

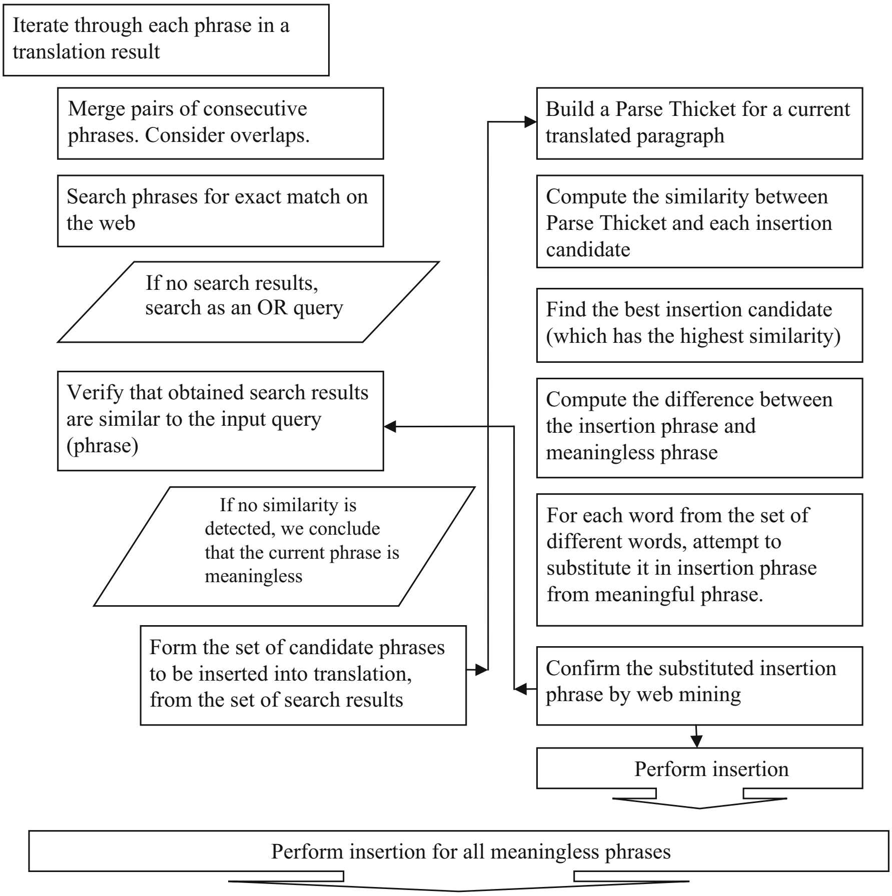

Fig. 12.12 Algorithm for posting correction via web mining

If the highest score is still below the threshold, we conclude that there is no document on the web with an expression similar to the one under consideration, and it is therefore *meaningless*. Otherwise, if a document with an expression similar to the one under consideration is found, we conclude that it is *meaningful* and proceed to the next expression from L_{op}. Our assumption here is that it is almost impossible to "invent" a new expression that does not exist on the web. Therefore the CASP posting correction system tries to find an existing 'analogue' from a trusted source for an "invented" expression in the translation result, believed to be meaningless, according to our model.

For the list of meaningful search results for a meaningless expression, we try to find which search result is the most appropriate. To do that, we generalize each search result with the whole translation test (Chap. 5). For that we build a parse thicket for the latter, and generalize it with parse tree of the former. Given a generalization score, we form the best insertion candidate from parts of the best search results (title, snippet, sentence).

Then we align the expression from L_{op} to be substituted and the best insertion candidate. For the former and for the latter, we form a set of pairs of words, which are different (does not occur in generalization of the expression from L_{op} to be substituted and the best insertion candidate). As a result of this step, we form a list of pairs of words, so that the word from the meaningful expression will be substitute by the word from the meaningless one. The more such substitution we perform, the closer the result will be to the original translation, but potentially more meaningless. Hence we want to verify that as a result, substituting words in meaningful expression, this expression remains meaningful.

Hence each substitution results need to be verified in the same way each expression from L_{op} is verified. Once all word substitutions are verified, the resultant expression can be substituted instead of the original member of L_{op}. Once this is done for each expression in L_{op}, the algorithm concludes.

12.10 More Examples of Conversations

In our first example, we have a housewife conversation about finding a costume for performance:

M: Uh, Helen, we don't need to go to a thrift store to make you look like a punk rocker for your next theater performance (Fame Jr.!). We have a variety of options right at home (although we could use suspenders and doc martens size 7 to pull off the red plaid skirt). The pink hair dye and we're good.

E: Ysa has skull and crossbones suspenders that Maddy can borrow. Sadly, all my Docs are too big for her. (but Ysa's been BEGGING for "combat boots"-is it the age?? I was NOT that cool at 10...or ever!)

M: Oh, yes to the suspenders! (My docs are too big too, and a bit dusty, but maybe WE can share.

L: I have some red creepers in size 5 (men's), if that's of interest.

CASP: I like punk rock. But I'm not a punk rocker. Growing up being a guitar player and being into music, there were really only two types of people you could hang out with: heavy metal dudes and punk rockers. They were the only two ... Our teacher, Linda Harlan, gave us an "A" not because it was good, but because we had been so ballsy in taking over the entire school to make our silly movie. ..... It occurred to me, how am I gonna drive home, I don't have my glasses!

This conversation is yielded by a posting photos included an album with a certain name:

G: added 7 new photos to the album Prague.
S: Shame about the weather.
R: I dont see any caving pics in here Gilly.
G: Guess I'm just sick of caves.. just kidding! I'm en route to caving.
CASP: Considering also, I had never been to Prague before (unless you count that one time I passed it by on the way to Hungary), I couldn't wait to count down the days till leaving for a short road trip. ... Afterwards we casually walked down tiny, narrow alleyways, checked out Náměstí Republiky, Wenceslas Square, stopped at Sephora's (tiniest Sephora I've ever been to and aside from NARS they didn't sell anything one wouldn't get in Germany anyway)

Although the host has never been to Prague (which is true) the rest of information has nothing to do with the host travel experience but nevertheless feels authentic. Now a Facebook user makes a post with a reference to an article in economics, so CASP needs to generalize from both texts to form a lattice query. The discussion is started to heating up with the participants asking and answering individually, as CASP joins. . .

E: I think we're moving to an era where companies and countries have to decide who they trust more: the United States (and US domiciled companies) or China (and Chinese companies like Huawei). I'm not saying it's an obvious choice either...but we may be seeing the beginning of the the Balkanization of not just the internet, but enterprise IT as well. [Huawei has spied for Chinese government, ex-CIA boss says www.guardian.co.uk. Michael Hayden, also former head of NSA, says he is aware of hard evidence of spying activity by world's No 2 telecoms firm . . .]
A: it's always a nice surprise to see you drop by MVODC. BTW, how do you know David Kim? He's the leader of my small group. Re: enterprise IT. I see what you mean now. I interpreted "enterprise" at the company level, but you were referring to "enterprise IT" in the whole, by country. Sure, I can see Balkanization, and trade treaties, and so on.
S: I suspect my new Samsung 'fridge is spying on me for the Korean government. It's just sitting there in the corner, like, "who, me? A spy? Are you kidding?"
K: It would be an obvious choice for me. El, you need to get back to the U.S.
E: @K yeah...I trust Google more than most other companies in the US or China...but other companies like MSFT YHOO etc probably would just give it all up to USG without much of a fight.

(continued)

CASP: The textile industry is very concerned that the TPP will specifically allow a backdoor for Vietnam (a TPP member country) to buy Chinese textiles and sell or integrate them as their own under the TPP. ...I totally understand that this will cost more, but I would rather trust the quality of American-made goods (Our QA-QC is worlds apart from Continental Asian companies... especially China) over those made in China.

Our next example is CASP acting in advertising mode (Fig. 12.13). For a posting '*I want to go to Lady Gaga 'Born This Way', and who wants her gigs stopped?*' CASP chatbot provides a web page with an option to purchase a tickets and also provides a text related to *who wants her gigs stopped.* Relevant images are provided as well. Since the seed posting is controversial, the chatbot tries to find a content from the both sides, Lady Gaga fans and religious groups worrying about how Lady Gaga affects the youth.

12.11 Discussion and Conclusions

We proposed a chatbot domain of social promotion and built a conversational agent CASP to act in this domain (Fig. 12.8). CASP maintains friendship and professional relationship by automatically posting messages on behalf of its human host, to impress the peers that the human host thinks and cares about them. Also, communicating issues raised by peers, CASP can be set to mention various products and services, providing implicit advertisement. We observed that a substantial intelligence in information retrieval, reasoning, and natural language-based relevance assessment is required so that members of the social network retain interest in communication with CASP. Otherwise, the result of social promotion would be negative and the host would loose friends instead of retaining them. We demonstrated that a properly designed social promotion chatbot could indeed relieve its human host from the efforts on casual chatting with her least important friends and professional contacts.

According to Buchegger and Datta (2009), online social networks are inherently peer-to-peer (P2P). Building them as P2P networks leverages a scalable architecture that can improve privacy and avoid the "big brother" effect of service providers. Moreover, Web search engines have problems providing good Web coverage, given the Web's size and high rates of change and growth. It can result in information overload (Wu et al. 2008; Galitsky et al. 2010). Furthermore, the most valuable information is not always available, as in the case of the deep Web. The deep Web is WWW content that is not accessible through search engines; its volume was estimated to be thousand times higher than the visible Web. Moreover, centralized horizontal search engines aim to satisfy the needs of any user type and they are

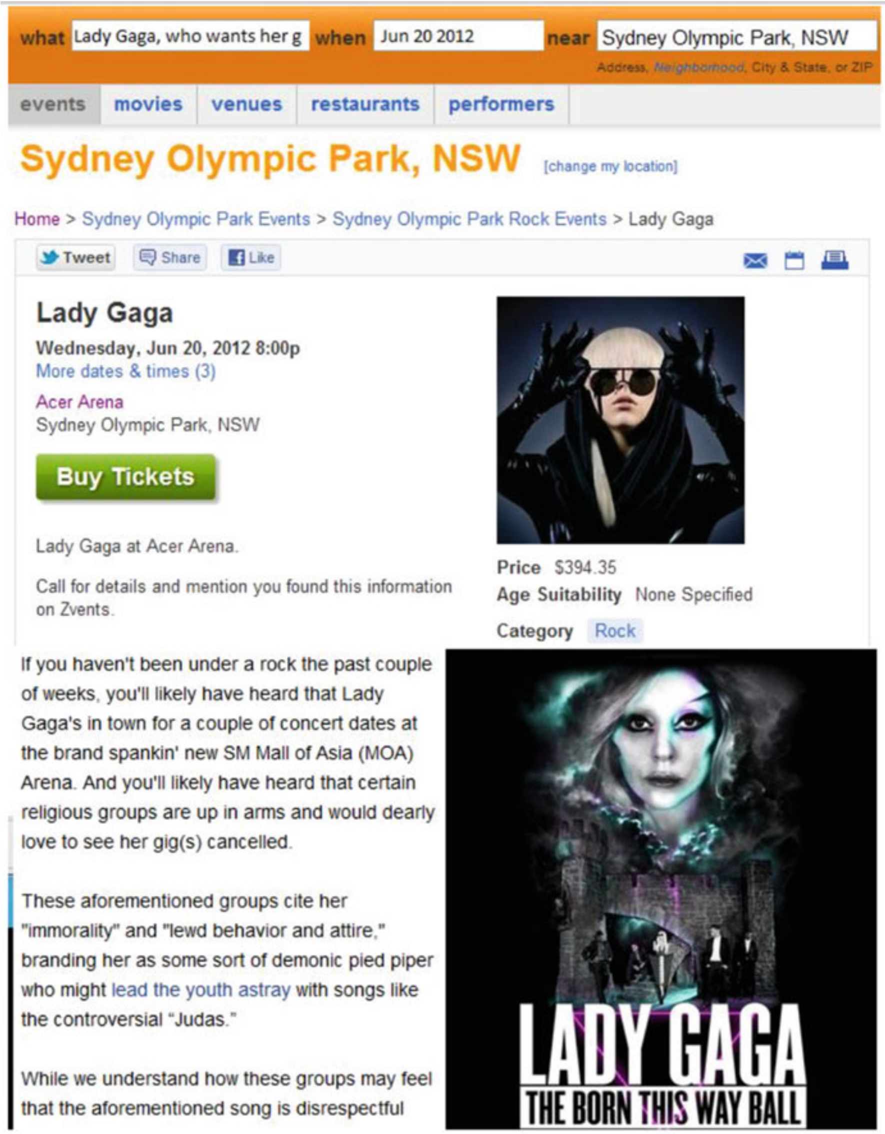

Fig. 12.13 CASP comments on a controversial topic related to a artist and also offers a web form to buy a ticket

progressively personalized and context aware; although they generally provide good results, they are less effective when dealing with atypical searches.

For the purpose of promoting social activity and enhance communications with the friends other than most close ones, the chatbot is authorized to comment on postings, images, videos, and other media. Given one or more sentence of user posting or image caption, CASP issues a web search request to Bing or an internal company resource and filters the search results for topical relevance, rhetoric appropriateness and style. Experiments with Facebook account were conducted using Facebook OpenGraph involving a number of personal accounts.

To extract a topic and form a query from a conversational thread, we introduced a new type of query for search engine framework, the lattice query, which is intended to facilitate the process of an abstract data exploration. Instead of having a user formulate a query, one or more instances are automatically formed from sample expressions. To derive a lattice query, as well as measure relevance of a question to an answer, an operation of syntactic generalization (Chap. 6, Galitsky 2014) is used. It finds a maximal common sub-trees between the parse trees for the sample text fragments, and also it finds the maximum common sub-trees between the parse trees for the lattice query and that of the candidate answers. In the latter case, the size of the common sub-trees is a measure of relevance for a given candidate search result.

In our evaluation we compared the conventional information extraction approach where extraction rules are expressed using keywords and regular expressions, with the one where rules are lattice queries. We observed that lattice queries improve both precision and recall of information extraction by producing more sensitive rules, compared to sample expressions which would serve as extraction rules otherwise. For the web search, if one wants to find information relevant to a few portions of text, such as blog postings, Facebook reply or couple of articles of interest, lattice queries are a handy tool. It forms a web search (lattice) query to find relevant results on the web and access their similarity. An importance of the lattice queries in data exploration is that only the most important keywords are submitted for web search, and neither single document nor keyword overlap deliver such the set of keywords.

We performed the evaluation of relevance assessment of the CASP web mining results and observed that using generalization of parse thickets for the seed and candidate message is adequate for posting messages on behalf of human users. Chatbot intelligence is achieved in CASP by integrating linguistic relevance based on parse thickets (PT, Chap. 7) and mental states relevance based on simulation of human attitudes (Galitsky 2016). As a result, CASP messages are trusted by human users in most cases, allowing CASPs to successfully conduct social promotion.

We experimentally observed the correlation between the intelligence components of CASP and peers' willingness to use it: once these components start to malfunction, the human users begin to complain and even intend to disconnect from CASP. In the human-human network, events when people unfriend their peers occur in case of a problem in their relationship, strong deviations in their beliefs and political opinions, but not when humans post least meaningful and least appropriate messages. Humans are ready to tolerate a lack of intelligence in what other humans write, in most of the cases. On the contrary, when chatbot utterances are irrelevant or inappropriate, the tolerance is not that high.

We tracked the usability scenarios of CASP when users ended up unfriending it and even encouraging others to do that, measuring topical and rhetoric relevance values, as well as the number of repetitions of problematic postings. We observed that CASP substantially outperforms the boundary area where a significant number of peers would avoid using it. It is confirmed that the events of unfriending happen rarely enough for CASP agent to improve the social visibility and maintain more friends for a human host than being without CASP. Hence although some friends lost trust in CASP, the friendship with most friends was retained by CASP; therefore, its overall impact on social activity is positive.

CASP was featured on BBC Inside Science (2014). “Keeping up with your online social network of ‘friends’ on Facebook can sometimes be time consuming and arduous. Now CASP is designed to do the bulk of his social interactions online. But how realistic is it? And does it fool his cyber pals?” – these were the questions of the reporter.

According to New Scientist (2014) article “Facebook for lazybones”, if one wants to stay in touch with friends on Facebook but cannot be bothered to do it himself, he should rely on CASP which monitors the social media feed and responds as if it is the host person. CASP makes relevant comments on photos posted by Facebook friends by analyzing the text of status updates and then searches the web for responses.

The content generation part of CASP was available at www.writel.co in 2014–2016. Given a topic, it first mined the web to auto build thesaurus of entities (Galitsky and Kuznetsov 2013, Chap. 8) which will be used in the future comment or essay. Then the system searches the web for these entities to create respective chapters for these entities. The resultant document is delivered as DOCX email attachment.

In the interactive mode, CASP can automatically compile texts from hundreds of sources to write an essay on the topic. If a user wants to share a comprehensive review, opinion on something, provide a thorough background, then this interactive mode should be used. As a result an essay is automatically written on the topic specified by a user, and published. The content generation part of CASP is available at www.facebook.com/RoughDraftEssay.

References

BBC Inside Science (2014) Automatic Facebook. http://www.bbc.co.uk/programmes/b040lnlf

Bhasker B, Srikumar K (2010) Recommender systems in e-commerce. CUP. ISBN 978-0-07-068067-8

Borgida ER, McGuinness DL (1996) Asking queries about frames. In: Proceedings of the 5th international conference on the principles of knowledge representation and reasoning, pp 340–349

Buchegger S, Datta A (2009) A case for P2P infrastructure for social networks – opportunities & challenges. In: Proceedings of 6th international conference on wireless on-demand network systems and services, Utah, pp 161–168

Buzmakov A (2015) Formal concept analysis and pattern structures for mining structured data. Inria Publication. https://hal.inria.fr/tel-01229062/
Cassell J, Bickmore T, Campbell L, Vilhjálmsson H, Yan H (2000) Human conversation as a system framework: designing embodied conversational agents. In: Cassell J et al (eds) Embodied conversational agents. MIT Press, Cambridge, MA, pp 29–63
Chambers N, Cer D, Grenager T, Hall D, Kiddon C, MacCartney M, de Marneffe C, Ramage D, Yeh E, Manning CD (2007) Learning alignments and leveraging natural logic. In: Proceedings of the ACL-07 workshop on textual entailment and paraphrasing
De Rosis F, Pelachaud C, Poggi I, Carofiglio V, de Carolis B (2003) From Greta's mind to her face: modeling the dynamics of affective states in a conversational embodied agent. Int J Hum Comput Stud 59:81–118
Dias J, Paiva A (2005) Feeling and reasoning: a computational model for emotional characters. In: EPIA affective computing workshop, Springer
Galitsky B (1998) Scenario synthesizer for the internet advertisement. Proc J Conf Infol Sci Duke Univ 3:197–200
Galitsky B (2003) Natural language question answering system: technique of semantic headers. Advanced Knowledge International, Adelaide
Galitsky B (2013) Transfer learning of syntactic structures for building taxonomies for search engines. Eng Appl Artif Intell 26(10):2504–2515
Galitsky B (2014) Learning parse structure of paragraphs and its applications in search. Eng Appl Artif Intell 32:160–184
Galitsky B (2016) Theory of mind engine. In: Computational autism. Springer, Cham
Galitsky B (2017) Content inversion for user searches and product recommendation systems and methods. US Patent 15150292
Galitsky B, Ilvovsky D (2016) Discovering disinformation: discourse-level approach. Fifteenth Russian national AI conference, Smolenks, Russia, pp 23–33
Galitsky B, Kovalerchuk B (2006) Mining the blogosphere for contributor's sentiments. AAAI Spring symposium on analyzing weblogs. Stanford, CA
Galitsky B, Kuznetsov SO (2008) Learning communicative actions of conflicting human agents. J Exp Theor Artif Intell 20(4):277–317
Galitsky B, Kuznetsov SO (2013) A web mining tool for assistance with creative writing. ECIR, European conference on information retrieval, pp 828–831
Galitsky B, Levene M (2007) Providing rating services and subscriptions with web portal infrastructures. Encyclopedia of portal technologies and applications, pp 855–862
Galitsky B, McKenna EW (2017) Sentiment extraction from consumer reviews for providing product recommendations. US Patent 9,646,078
Galitsky B, Parnis A (2017) How children with autism and machines learn to interact. In: Autonomy and artificial intelligence: a threat or savior? Springer, Cham, pp 195–226
Galitsky B, Shpitsberg I (2016) Autistic learning and cognition. In: Computational autism. Springer, Cham, pp 245–293
Galitsky B, Tumarkina I (2004) Justification of customer complaints using emotional states and mental actions. FLAIRS conference, Miami, Florida
Galitsky B, Usikov D (2008) Programming spatial algorithms in natural language. AAAI workshop technical report WS-08-11, Palo Alto, pp 16–24
Galitsky B, Kuznetsov SO, Samokhin MV (2005) Analyzing conflicts with concept-based learning. Int Conf Concept Struct 3596:307–322
Galitsky B, Dobrocsi G, de la Rosa JL, Kuznetsov SO (2010) From generalization of syntactic parse trees to conceptual graphs. In: Croitoru M, Ferré S, Lukose D (eds) Conceptual structures: from information to intelligence, 18th international conference on conceptual structures, ICCS 2010, Lecture notes in artificial intelligence, vol 6208, pp 185–190
Galitsky B, Dobrocsi G, de la Rosa JL, Kuznetsov SO (2011) Using generalization of syntactic parse trees for taxonomy capture on the web. 19th international conference on conceptual structures, pp 104–117

Galitsky B, de la Rosa JL, Dobrocsi G (2012) Inferring the semantic properties of sentences by mining syntactic parse trees. Data Knowl Eng 81–82(Nov):21–45

Galitsky B, Ilvovsky D, Kuznetsov SO, Strok F (2013) Finding maximal common sub-parse thickets for multi-sentence search. IJCAI workshop on graphs and knowledge representation, IJCAI 2013

Galitsky B, Ilvovsky D, Lebedeva N, Usikov D (2014) Improving trust in automation of social promotion . AAAI Spring symposium on the intersection of robust intelligence and trust in autonomous systems, Stanford, CA, 2014

Jiang JJ, Conrath DW (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the international conference on research in computational linguistics, Taiwan

Kushman N, Artzi Y, Zettlemoyer L, Barzilay R (2014) Learning to automatically solve algebra word problems. ACL 2014

Lawless WF, Llinas J, Mittu R, Sofge DA, Sibley C, Coyne J, Russell S (2013) Robust intelligence (RI) under uncertainty: mathematical and conceptual foundations of autonomous hybrid (human-machine-robot) teams, organizations and systems. Struct Dyn 6(2):1–35

Lisetti CL (2008). Embodied conversational agents for psychotherapy. CHI 2008 workshop on technology in mental health, New York

MacCartney B, Galley M, Manning CD (2008) A phrase-based alignment model for natural language inference. The conference on empirical methods in natural language processing (EMNLP-08), Honolulu, HI, October 2008

Makhalova T, Ilvovsky DI, Galitsky B (2015) Pattern structures for news clustering. FCA4AI@ IJCAI, pp 35–42

Montaner M, Lopez B, de la Rosa JL (2003) A taxonomy of recommender agents on the internet. Artif Intell Rev 19(4):285–330

New Scientist (2014) http://www.newscientist.com/article/mg22229634.400-one-per-cent.html

Ourioupina O, Galitsky B (2001) Application of default reasoning to semantic processing under question-answering. DIMACS Tech Report 16

Reeves B, Nass C (1996) The media equation: how people treat computers, television, and new media like real people and places. Cambridge University Press, UK

Sidorov G, Velasquez F, Stamatatos E, Gelbukh A, Chanona-Hernández L (2012) Syntactic N-grams as machine learning features for natural language processing. Expert Syst Appl 41 (3):853C860

Strok F, Galitsky B, Ilvovsky D, Kuznetsov SO (2014) Pattern structure projections for learning discourse structures. AIMSA 2014: artificial intelligence: methodology, systems, and applications, pp 254–260

Trias i Mansilla A, de la Rosa i Esteva JL (2011) Asknext: an agent protocol for social search. Inf Sci 2011:186–197

Trias AJL, de la Rosa B, Galitsky G (2010) Drobocsi, automation of social networks with QA agents (extended abstract). In: van der Hoek W, Kaminka GA, Lespérance Y, Luck M, Sen S (eds) Proceedings of 9th international conference on autonomous agents and multi-agent systems, AAMAS ‘10, Toronto, pp 1437–1438

Wu LS, Akavipat R, Maguitman A, Menczer F (2008) Adaptive peer to peer social networks for distributed content based web search. In: Social information retrieval systems: emergent technologies and applications for searching the web effectively. IGI Global, Hershey, pp 155–178