Introducing `Consciousnet` : Internet Content as an Environment for Human-Machine Interaction

Vincenzo Catania[1], Davide Patti[1], and Mariagrazia Sciacca[2]

[1] DIEEI, University of Catania, Italy
{vincenzo.catania,davide.patti}@dieei.unict.it
[2] Dipartimento Scienze della Formazione, University of Catania, Italy

Abstract. In this work we introduce `Consciousnet` , an open source architecture aimed to provide a general purpose environment for experimenting with human-machine language interaction. The main idea is exploiting the distributed and unsupervised complexity of the Internet in order to get all the semantic/syntactic material needed to carry on a linguistic text based interaction. After describing the main elements of the architecture, the results of a set of Turing-inspired tests are shown to demonstrate how the unpredictability and generality of the environment can be used as a basis for designing tests and experiments involving both psychologists and AI scientists.

1 Introduction and Motivation

Language-based interaction between human and machines has always attracted several actors belonging to very heterogeneous fields, from computer architecture designers to cognitive science researchers, psychologists, language formalists, philosophers and sometimes also artists [6] [11] [12] [2]. While the final purpose behind each of these fields may be different, what all the approaches have in common is the intrinsic difficulty of dealing with a language-based interaction. Language is still probably the most hardly-reproducible behaviour of human entities, strictly related to the inner complexity of the way human intelligence represents and interacts with the environment. If several physical features of the human body have been mechanically replicated in the recent years [20] [4], the same success did not come for language-based interaction, where the hunt for a more human-like behaviour is still wide open.

In this work we present `Consciousnet` , an artificial intelligence environment for experimenting with human-computer linguistic interaction. The name itself comes from the contamination of the words *consciousness* and *net*, denoting the main idea which characterizes the environment: exploiting the chaotic, unsupervised knowledge of the Internet as a collective "consciousness" that can be stimulated by the user while interacting with an artificial entity.

Three fundamental requirements are behind the design philosophy adopted within `Consciousnet` :

Pure Text-Based Interaction: no need for any added "realism" based on multimodal techniques. This excludes speech synthesis/recognition, three-dimensional

M. Kurosu (Ed.): Human-Computer Interaction, Part II, HCII 2014, LNCS 8511, pp. 382–393, 2014.

avatars, touch-based interactions or robotics. The idea was focusing on linguistic interaction instead of introducing distracting elements revealing/recalling the artificial nature of the interaction. For example, although speech synthesis is widely used to "humanize" the user experience, it makes the interaction more recognizable as non-human, while text-based output are more maskable.

Generality: not specialized or focused on a particular conceptual domain. So, while limited to a pure text-based interaction, the space of action of that interaction should have no limitation a priori. This, for example, differentiates `Consciousnet` from the field of expert systems and assistance or entertainment chat robots.

Unstructured Complexity: only simple and easily adaptable components should be used, with the aim to generate complexity from their interaction rather than forcing them to behave in a complex way. Thus we also avoided the usage of semantic/formal systems aimed to capture the "meaning" of the user input. In other words, the complexity is not encoded *inside* the functional model of `Consciousnet` , but is obtained by stimulating the intrinsic complexity of the Internet content (see Figure 1).

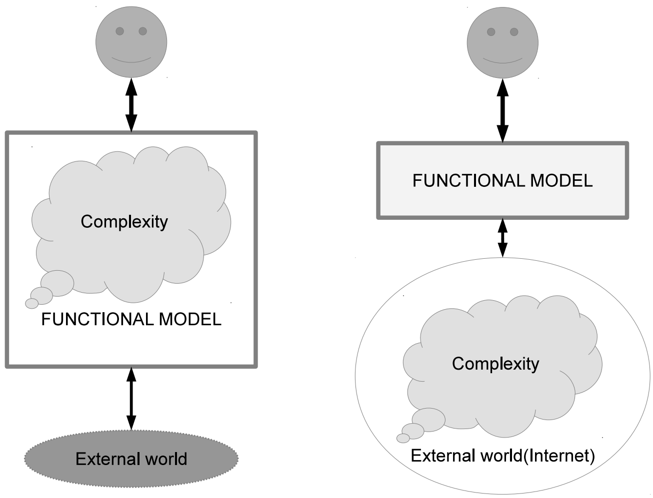


Fig. 1. Structured vs unstructured approach adopted in `Consciousnet`

2 Background and Contribution

From a computer science/software perspective, `Consciousnet` could be classified as a *chatbot*, that is a program which simulates an artificial intelligence capable to textually interact with users. This field, originated by the seminal work [19], was followed by many implementations over the years [3] [16] [17], basically differentiating each other by two main aspects: (i) the complexity of the parsing model applied to the user input and (ii) the knowledge base used to generate responses. In the last year some works also proposed to feed this knowledge base using Internet resources, e.g. discussion forums [9] [1].

The contribution we aim to introduce with `Consciousnet` is the attempt to remove any explicit knowledge base and use the Internet as an autonomous structure which provides the semantic and syntactic material that can be forged to create the interaction. In particular, two aspects we want to emphasize here:

Unsupervised linguistic space: no database of concepts or archive of responses to be maintained; the current status of the Internet itself determines the size and the content of the space into which `Consciousnet` moves. It should be pointed out that this affects both semantics and syntactical aspects of this space (e.g. slangs, abbreviations, common errors are part of this linguistic space)

Unpredictability: the same nature of Internet content, fluid, mutable, intrinsically chaotic and hardly controllable leads to an interesting degree of indeterminism in the behaviour of `Consciousnet` . Even knowing the user inputs in advance, it would be hard to predict the interaction development.

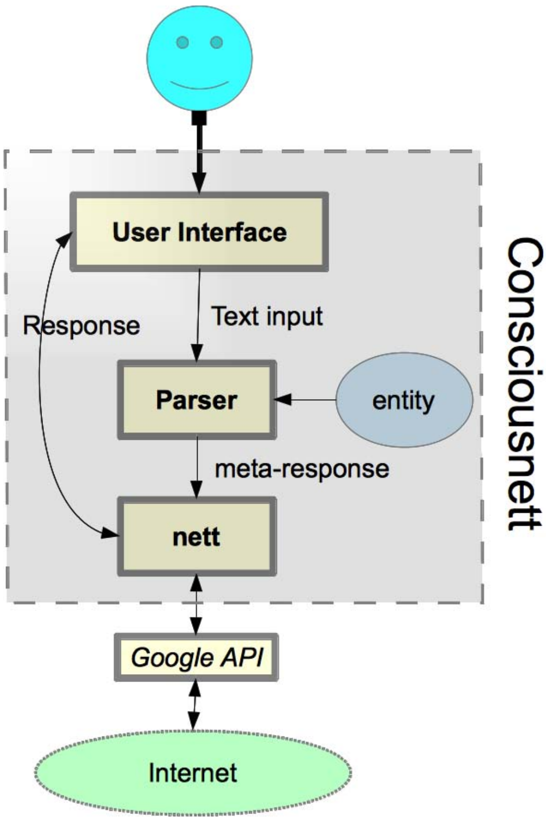


Fig. 2. `Consciousnet` Architecture components and data flow

3 Consciousnet Architecture

In this section we introduce the architecture of the Consciousnet environment and a detailed description of its elements. The source code of the whole environment is freely available at [15], together with the appropriate instruction for setting up the environment and the complete set of experiments carried out in the next Section.

3.1 Architecture Sketch

As shown in Figure 2, the text input is introduced by the user using an User Interface and then analysed by the *Parser* in order to produce a *meta-response* . The *meta-response* is not the final output of the entity, but some kind of "reaction" that will be used by the subsequent *Net* component to stimulate the Internet. The actual interface between *Net* and the Internet consists of a set Google APIs [10], freely available for non-commercial purposes. Once the *Net* component has processed the output of these API, the final response is returned to the user so that the interaction loop can repeat. As shown, a data structure called *Attitude* is used to generate the *meta-response* : *Attitude* represents the controllable part of the entity behaviour, in some way determining its "personality". In the following subsection we describe more in detail each component involved in the interaction loop.

3.2 User Interface

While the interaction is simply based on text input/output, a few tricks have been adopted in order to make the environment suitable for a more realistic experience. First, the entity response is not returned immediately after user input is entered, but following a time delay $T_{response\_delay} = T_{read} + T_{think}$, where both values are proportional to the number of chars of the text strings involved, with the aim of simulating the time required for reading the user input and then thinking a response. Further, text does not appears on the interface screen all at once, but as a flow of randomly intervalled chunks of chars, like happening in a live typing session. This last trick was not really necessary as the previous one, but adding some more human-like typing to the remote entity demonstrated to mitigate the artificiality intrinsic in the proportional delay $T_{response\_delay}$.

Note that all of these tricks could have been simply avoided using an hidden human counterpart in order to type the output generated by the artificial entity. However, having a self-contained environment which includes a modelization of controllable human-like typing/reading yields a further degree of flexibility of the environment as experimental platform, e.g. different delay values could be investigated to evaluate the effect of slower/faster interactions on language.

3.3 Parser

As next step, input is analysed by the *Parser* using the *Attitude* data structure in order to generate a *meta-response* . The *Attitude* consists in a hierarchy of *entry-point* , *decomposition pattern* and *meta-response* , as follows:

```
entry-point: X
   pattern: X1
       meta-response X1.1
       meta-response X1.2
       meta-response X1.3
   pattern: X2
       meta-response X2.1
       meta-response X2.2
       meta-response X2.3
```

The *entry-point* represents a sort of keyword that opens the understanding of the input. The idea is to think to the artificial entity like a person trying to understand some sentences in a foreign language. The first thing should be to capture as more words as possible that provide a meaningful interpretation key for the whole sentence. Of course, more than one these entry points could be found in each input, so a sort of priority mechanism has been chosen. Continuing the analogy with the foreign language, we can observe that the more abstract and general a word is, less is the semantic value useful as understanding entry point for the whole sentence. For example, denoting with "*" the not understood parts of a sentence, catching a pattern like "* *me* * *" does not help, since abstract words like "me", "you", "is","are" are very common and do not carry any particular semantic weight to characterize the meaning of the sentence. A less abstract entry-point, e.g. "food", could be more useful in that sense, since one could at least argue that the counterpart is talking about something to eat. Very low abstract entry-points, e.g. the name of a city or a car model, could give even more hints when trying to build a conversation in a foreign language, since you can have a more accurate understanding of which conceptual domains could be touched from the current stage of the dialogue. Table 1 show the class of entry points considered, ordered by abstraction level.

Once the *Parser* has used the *Attitude* data structure to find the *entry-point* with the highest priority (lowest abstraction), a set of decomposition pattern is

Table 1. entry-point list and abstraction levels

Level	Type of Element	Examples
0	Commonly found, generic syntactical elements, i.e. not useful for any restriction of the conceptual domains	I, me, are, you, sorry, yes, no
1	Verbs, nouns, elements denoting something less generic, such hypothesis, questions etc...	if, because, why, how, when, always
2	Terms introducing items that assume importance to the speaker	my X, your X
10	Terms introducing specific domains	Music, sport, love, school, food, money
10+	Very low level abstraction terms, referring to a very specific subject	Mozart, Golf, Berlin, spaghetti

considered in order to determine the constituting elements of the sentence and build an appropriate *meta-response* . As a practical example, let's consider the following snippet of *Attitude* . For sake of simplicity, it's a very basilar pattern with only a few entries:

```
entry-point: love
    pattern: * I love *
        meta-response: fans of (2)
        meta-response: (1) because loving (2)
    pattern: * when *
        meta-response: goto when
    pattern: * love *
        meta-response: (1) hate (2)
entry-point: when
    pattern: * when *
        meta-response: how often (2)
```

This example shows two *entry-point* : the first is associated to the word "love" and has three patterns. For each pattern, a set of *meta-response* is available. The input is compared against matching patterns, extracting some placeholders (e.g. (1) and (2) in the example), and then translated into a *meta-response* . The concept of *meta-response* is probably the most important in the architecture. As said, it is not the final output of the artificial entity, but an input that will be used by the *Net* component to generate the actual response. In this example, an user input like "In the morning I love cats" would match only the first pattern, mapping the placeholders (1) and (2) to "In the morning" and "cats" respectively. A randomly chosen item in the corresponding set of meta-responses (e.g. "fans of cats") will be then used by the subsequent *Net* component to access the Internet. Note how is possible to use one of the pattern to delegate the *meta-response* to a different *entry-point* ("when") if the corresponding keyword is present. The idea is to use an *entry-point* with lower priority if no better choice is available. The set of *entry-point* is read in order, so in the example above the input "when I love cats" will match the first pattern, while "you love me when I play golf" would match the second, linking then to *entry-point* "when" and generating the *meta-response* "how often I play golf".

3.4 Net

The *Net* component implements an interface to the Google CustomSearch APIs [10], used in conjunction with the *meta-response* to extract data from the Internet and generating the actual response of the artificial entity. This involves the following phases:

1. The *Net* component uses its own CustomSearch Engine object using the *meta-response* as main argument.
2. As result, an array data structure containing fragments of data extracted from the internet is returned.

3. At this point, *Net* extracts the *snippet* field of the returned structure. Indeed, our aim is not to deal with low level web code (e.g. HTML or javascript), but with already human-readable text.
4. The resulting blob of data is then processed using an regular expression system implemented in *Net* , performing some post processing tasks: discarding too long/short sentences, strange punctuation, text with not useful content (e.g. all numbers).
5. Finally, each of the filtered items is given a sort of “quality value”, depending on some properties of the text, e.g. containing a first-person statement, having a question mark a last character and so on.

3.5 `Consciousnet` Tuning

Each of the components described is designed and implemented in order to work as a separate functional element. The degrees of freedom in the environment configuration are mainly encoded in the *Attitude* data structure. A lot of flexibility comes from tuning the “personality” of the entity in order to accomplish some specific tasks. We can distinguish between two different ways of tuning:

– *Extensively*: this consist in adding new *entry-point* items to the *Attitude* structure, expanding the conceptual domains that can be used as starting point for creating the *meta-response*
– *Intensively*: for a given *entry-point* , the number of decomposition patterns and *meta-response* could be customized in order to enrich the complexity of the behaviour mapped in each *entry-point* .

Users oriented toward an “expert-system” perspective would probably tune `Consciousnet` extensively, adding several *entry-point* elements focused on a particular domain, for example name of symptoms and diseases in an health-care application. Further, the complexity of *meta-response* could also be limited in order to get simpler and controllable text snippets; we cannot exclude interesting results in these ways of tuning `Consciousnet` , but they are apart from the original spirit of the environment and will not further discussed in this introductory work.

On the other side, different fields such as entertainment, language research, psychologists, could benefit from the flexibility of the environment and use the unpredictability and generality as an added value. In the interaction experiments between students and Prof. Gioio entity, presented later in this work, we used an *Attitude* including very heterogenous patterns, e.g. where `I love (1)` could result in the *meta-response* `Reasons why I hate (2)`, generating the not-so-friendly exchanges like:

```
You> I love studying psychology, hope will do good
Entity> I'd like to know from people studying psychology, if
they think
```

The choice of accessing external Internet complexity using appropriate *meta-response* instead of large databases avoids problems of scalability and maintainance that would arise when dealing with a not-specialized, general purpose system. For example, considering the snippet:

```
entry-point: play
   pattern:   I * play *
      meta-response:  people that (1) play (2)
      meta-response:  is the best for (2)
```

One of the experiments performed in the next Section, showed the exchange:

```
You> I used to play bass guitar
Entity> For distortion, definitely the Blackstar HT-Dual
```

which is really satisfying as interaction, even if not technical data about guitar tools has ever been introduced in Consciousnet . It has been sufficient to provide a good pattern to catch the placeholder "bass guitar", without really encoding anything strictly related to this particular instrument. In other words, the same patterns could give interesting results even when playing "drums" or "lego building blocks". As seen in Figure 1, a fundamental design choice is to leave complexity outside the environment enabling intelligent ways of letting it emerge from the network. As a further interesting consequence, if something that does not exists today will appear tomorrow, the collective consciousness of the network will be absorbing it, and then it will become something that Consciousnet could successfully use in its dialogues.

4 Example: Reversed Turing Test

In order to demonstrate the effectiveness of Consciousnet flexibility, this section shows how the environment has been used to carry out a set of Turing-test inspired experiments, specifically designed for this work. The idea was investigating some properties of the language used in two different sets of users: a set α, being aware of the artificial nature of the entity and a second set β, not being aware of. While the original Turing's test consists of an human entity which analyses language to guess whether the counterpart is artificial or not, the proposed experiment completely reverses the perspective: we explicitly make statement about the other entity nature, analysing how language is affected from this awareness (see Figure 3).

4.1 Experimental Setup

The two sets α and β consisted of 30 students, taken from a course on Fundamental of Informatics for Psychology, held at University of Catania in 2013 [14]. Students belonging to the set α were instructed as they were performing a text only connection with an american Professor, Doctor Paul Gioio [1], interested into

[1] Named after the italian nickname of one of the authors' son.

testing a new form of interaction to be used in future on his own students. Each student entered (one per time) in a isolated room, where a 10 minutes chatting session was performed. In order to maximize the chance of masquerading the artificial nature of the entity, students who already performed the session were moved into separated room. This avoided the change of influencing successive students with doubts or considerations on what happened during their interaction. Of course, interaction sessions belonging to set β did not require such expedients and they were explicitly told to perform a chat with an artificial intelligence entity.

4.2 Results

When all the students of set α ended their 10 minutes session, the experiment was revealed. Quantifying "how much" Prof. Gioio was considered as real is not in the purposes of the experiments and it would be really difficult to gather such a measure: the majority of the students declared themselves as "surprised" and a few of them told of having developed some suspects. In every case, we are not interested in what is their opinion *after* the interaction, but how they acted *during* the interaction. Thus we can safely assume that even more suspicious students have been interacting supposing an human counterpart, since the short time available and the particularity of the situation forced them to adopt a conservative behaviour.

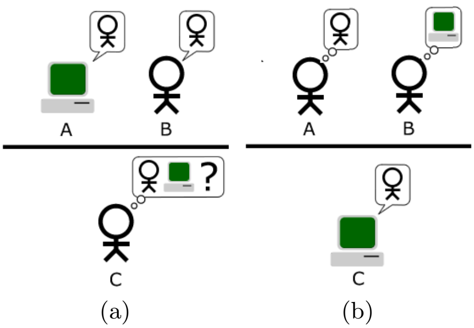


Fig. 3. Original Turing Test (a) and the reversed version (b) performed with `Consciousnet`

A total of 288 and 261 sentences have been collected from participants of sets α and β respectively, i.e. excluding those produced by the `Consciousnet` platform when interacting. A quantitative and qualitative analysis of the transcripts was conducted in order to investigate an impact in terms of semantic domains covered, syntax-oriented metrics, and statistically significative differences among the two sets α and β. Two kind of words have been removed from the collected sentence in order two produce less noisy results: the first set of words are those below a frequency threshold of 1% on the overall data. The second kind of words

Table 2. Most used words from Stanford POS analysis:α (top) and β (bottom)

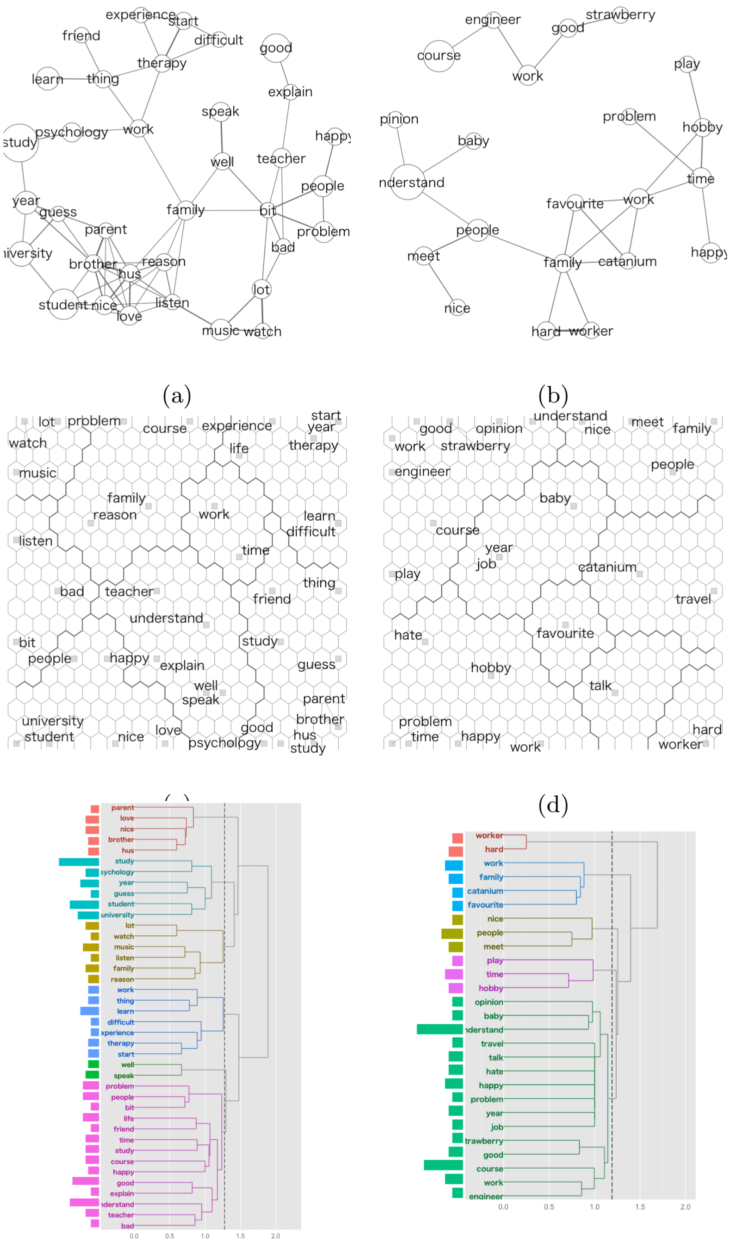
Noun		**Adj**		**Adv**		**Verb**	
study	15	good	10	well	4	understand	11
student	11	nice	5	exactly	2	learn	7
university	8	happy	4	probabily	2	love	5
year	7	bad	3	absolutely	1	speak	5
life	6	difficult	3	extremely	1	study	5
music	6	easier	2	frequently	1	start	4
people	6	favourite	2	good	1	work	4
Noun		**Adj**		**Adv**		**Verb**	
course	11	happy	5	hard	3	understand	13
people	6	good	4	well	2	work	5
time	5	hard	4	close	1	hate	4
work	5	favourite	3	dear	1	meet	4
family	4	nice	3	realy	1	talk	4
hobby	4	afraid	2	simply	1	play	3

excluded are those commonly referred as "stop words", i.e. words that do not play any particular semantic role (for a complete list of the stop words adopted, see also [13]).

Table 2 shows frequency list of most used words, labeled with the Stanford POS Tagger [7] considering 4 categories (noun, adjectives, adverbs, verbs). Further, a more complex analysis was conducted using the *R* [5] statistical tool in conjuction with *KH coder* [8], an open source software for content analysis, text mining or corpus linguistics. Part of the results obtained are summarized in the Figure 4, including co-occurrence network, self-organizing maps and clustering using method Ward with a Jaccard distance [18]. While these preliminary results seems to show some interesting differences between the two sets, any detailed and meaningful interpretation of such data is beyond the scope of this introductory word, which has been explicitly focused on the architecture of the `Consciousnet` environment.

5 Conclusions

In this paper we introduced `Consciousnet` , an artificial intelligence environment exploiting the Internet to perform a general purpose, not-specialized text based interaction. A set of experiments have been carried out to demonstrate how the environment has been used to perform a reversed version of the Turing test. Future works will involve both the improvement of the `Consciousnet` network-based intelligence and the design of new experimental tests for the research in the human-machine interaction field.



(a) (b) (c) (d) (e) (f)

Fig. 4. (a,b) Co-occurrence Network, (c,d) self-organizing maps and (e,f) word clustering for sets α and β

References

1. Cao, Y., Yang, W.-Y., Lin, C.-Y., Yu, Y.: A structural support vector method for extracting contexts and answers of questions from online forums. Inf. Process. Manage. 47(6), 886–898 (2011)
2. Clarke, A.C.: 2001: A space odissey. Pearson Education (2001)
3. Colby, K.M.: Simulation of belief systems. In: Computer Models of Thought and Language, pp. 251–286 (1973)
4. Eaton, M.: An approach to the synthesis of humanoid robot dance using non-interactive evolutionary techniques. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3305–3309. IEEE (2013)
5. The R Foundation for Statistical Computing. The r project for statistical computing, `http://www.r-project.org`
6. Goldstein, I., Papert, S.: Artificial intelligence, language, and the study of knowledge. Cognitive Science 1(1), 84–123 (1977)
7. The Stanford Natural Language Processing Group. Stanford log-linear part-of-speech tagger, `http://nlp.stanford.edu/downloads/tagger.shtml`
8. Higuchi, K.: Kh coder, `http://khc.sourceforge.net/en/`
9. Huang, J., Zhou, M., Yang, D.: Extracting chatbot knowledge from online discussion forums. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI 2007, pp. 423–428. Morgan Kaufmann Publishers Inc., San Francisco (2007)
10. Google Inc. Google customsearch, `https://developers.google.com/custom-search/`
11. Isles, D.: Artificial intelligence as a possible tool for discovering laws of logic. Cognitive Science 2(4), 329–360 (1978)
12. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. Stanford University (1968)
13. Ranks nl webmaster tools. English stopwords, `http://www.ranks.nl/resources/stopwords.html`
14. University of Catania. Dipartimento di scienze della formazione, `http://www.disfor.unict.it`
15. Patti, D.: Network knowledge based ai entity, `https://code.google.com/p/consciousnet/`
16. Shawar, B.A., Atwell, E.S.: Using corpora in machine-learning chatbot systems. International Journal of Corpus Linguistics 10(4), 489–516 (2005)
17. Tarau, P., Figa, E.: Knowledge-based conversational agents and virtual storytelling. In: Proceedings of the 2004 ACM Symposium on Applied Computing, SAC 2004, pp. 39–44. ACM, New York (2004)
18. Ward Jr., J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association 58, 236–244 (1963)
19. Weizenbaum, J.: Eliza a computer program for the study of natural language communication between man and machine. Commun. ACM 9(1), 36–45 (1966)
20. Yamane, K., Nakamura, Y.: Robot kinematics and dynamics for modeling the human body. In: Kaneko, M., Nakamura, Y. (eds.) Robotics Research. STAR, vol. 66, pp. 49–60. Springer, Heidelberg (2010)