Multimodal annotation scheme for online written conversation analysis

March 9, 2015

Contents

1 Introduction

1.1 Objects and definitions

The objects of our study are computer-mediated - or "online" - written conversations. This includes conversations taking place through a number of mediums, such as web forums, message boards, mailing lists, chatrooms, and website comment sections. It excludes all forms of oral speech, as well as their transcripts. It also excludes all forms of offline written speech, such as the dialogues between characters in a book, journalistic interviews or messages written in a guestbook.

1.1.1 Participants, messages and subjects

The conversations as we define them have three main characteristics:

- Multiple *participants*: a conversation is interactive by definition, and we define participants as human beings as well as bots taking actively part in the conversation by contributing new content and alterning between the roles of speaker and addressee.

- Multiple *messages*: a conversation is built of several messages from various participants, and we define messages as the smallest technical unit of communication handled by the given medium (e.g. in a forum, posts are messages, in a mailing list, emails are messages, etc.).

- A *subject*: conversations revolves around defined subjects, which are *not* necessarily the formally defined subject lines or forum thread names (easily extracted from metadata), but rather the underlying motivation behind the conversation: i.e., what it is *about*.

1.1.2 Dialogues, polylogues and roles

There is an important distinction between *dialogues*, conversations between two participants, and *polylogues*, conversations between two participants *or more*.

Usually, in corpora constituted exclusively of dialogues, each participant has a well defined *role*. For example, one may be a customer service agent and the other one a customer. In these scenarios, roles are static and remain the same through many conversations. Moreover, these conversations are often private: no one can "jump in" and send messages. But in other, more general corpora, where anyone can participate in a conversation, roles are

not as well defined and can fluctuate during a conversation. It is however true in most cases that the conversation initiator (i.e. the first participant to send a message) is likely to be the one with the problem and thus the only one who can define it, acknowledge solutions for it and actually act on it in the real world.

As of now, the annotation scheme does not explicitly disinguish one from the other. From this point on in this document, we use the word "dialogue" to refer to both kinds of conversations.

1.1.3 Problems and solutions

We are more specifically interested in a more particular kind of conversation: functional conversations, i.e. conversations that are designed to convey information in order to help achieve an goal. When the motivation behind a conversation is to find solutions to a stated problem, we consider them to be problem/solution oriented conversations. They have a clearly stated goal and can be *resolved*. These are at the heart of our research goals.

We define problems and solutions as follows:

- Problem: an unwelcome matter or situation affecting one or several participants, that is considered harmful or needs to be resolved or worked around.

- Solution: a way of solving or working around a specific problem previously introduced in the conversation, usually in the first message of the thread.

1.2 Goals

Our goal is to develop an annotation scheme and protocol for the classification of message fragments in problem/solution oriented online conversations. The scheme should facilitate the problem/solution oriented analysis of message text, which, in turn, can be used for a number of applications:

- Problem/solution oriented information retrieval (e.g. finding relevant documentation for a problem stated in natural language)

- Solution finding (e.g. the proposal of previous solutions that worked for a similar problem)

- Cross-modal analysis (e.g. using data from different sources to improve the efficiency of other tasks)

- Modality/channel relevance estimation (e.g. assisting users in finding the best place to find help)

- Etc.

Such a scheme would need to be:

- Multimodal: applicable to different online mediums

- Exhaustive: capable of covering the entire text of a message

- Unambiguous: annotation should be made simple by following a decision tree

- Problem/solution oriented: useful in modelizing problems and their proposed solutions

A further objective is to use such annotated data to produce a sample segmentation of our email corpus so that we can test the validity of the hypotheses made in [Hernandez and Salim, 2014].
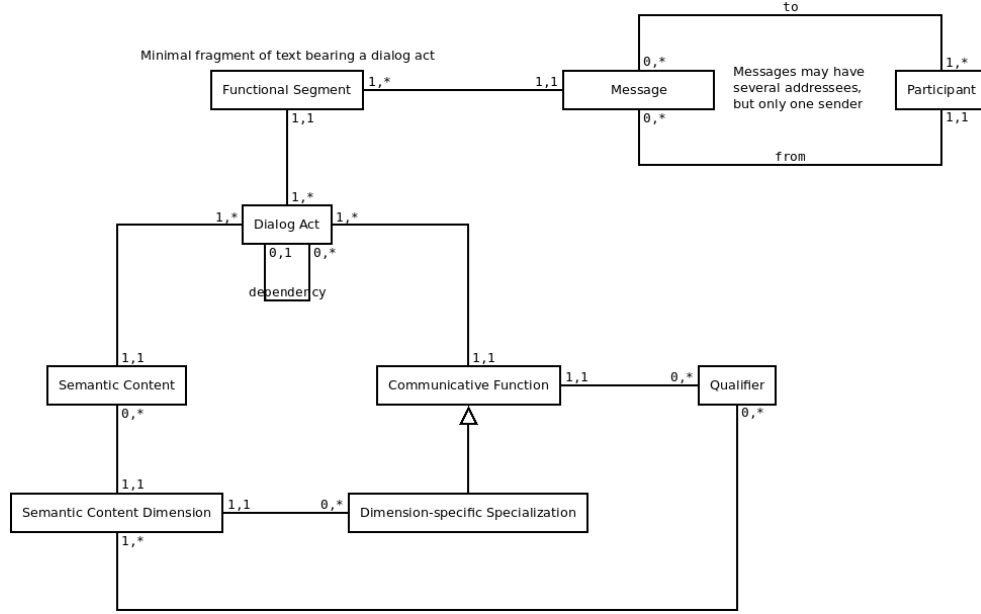
Figure 1: Meta-model

2   Proposal

This taxonomy of dialogue acts is derived from DIT++[Bunt, 2009].

2.1   Concept definitions

2.1.1   Dialogue acts

In the "information-state update" (also know as "context-change") approach to dialogue analysis, dialogue acts are interpreted as update operations applied to the information states of the interacting participants [Traum and Larsson, 2003, Bunt, 2011]. In this perspective, they are defined here as the conjunction of two components: a *semantic content* and a *communicative function* (see figure **??**).

The former, the semantic content, specifies the objects, propositions, and all the things that the dialogue act is about, and therefore contains all the information used in the update operation. The latter, the communicative function, specifies the way the dialogue act is intended to impact the information state of the addressee. For example, the utterance "do you know

4

what time it is?" is about the time that it is - that's its semantic content
- but its communicative function could be either a genuine question (the
speaker doesn't know the time but wants to find out) or a reproach (the
speaker noticed the addressee is late and is upset about it).

A functional segment is defined as the minimal fragment of text bearing
a dialogue act and is hereafter used interchangeably with the term "utter-
ance".

2.1.2 Dimensions in dialogue act annotation

Human communication is a complex activity. In functional conversations,
there is often a certain activity or task which the participants want to per-
form through the dialogue and as a result is the focus most interactions. In
conversations bearing requests for assistance, this task is to solve problems.
However, people do not communicate only to reach the formal objectives
of the conversation, they also constantly monitor the communicative pro-
cess, synchronize their mutual understanding through feedback utterances,
respect social conventions such as greeting and thanking, etc. Often, ut-
terances can be multifunctional and may serve several purposes related to
several of these things. For example, in the following exchange:

1. P1: Comment est ce que j'installe Dia ?

2. P2: Tout simplement : apt-get install dia

3. P1: Ça me dit permission non accordée je fais quoi ?

4. P3: Essaye avec sudo devant la commande

5. P1: C'est bon merci !

In the third utterance, participant 1 does two things: firstly, he informs
his audience that he ran into an issue, and, secondly, he asks for further
instructions. In the fifth utterance, he does two things again: he signifies
that his problem is solved, and he also thanks the participants that helped
him.

Such multi-functionality implies that accurate annotation of utterances
with dialogue act information calls for the assignment of more than one
tag to an utterance. This process is called "multi-label annotation". How-
ever in most multi-label schemes, only a small percentage of possible tag
combinations are actually used, due to the fact that many tags are in fact

mutually exclusive. This is an issue because it makes both human annotation and machine classification harder by unnecessarily increasing the task's complexity.

Here, like in DIT++, we guide annotators and classification algorithms in not considering impossible label combinations by the use of a multidimensional annotation scheme [Bunt, 2006]. The scheme allows each utterance to be annotated with one of 14 mutually exclusive tags per *semantic dimension*. Semantic dimensions are defined as the different areas within which dialogue acts can be semantically categorized.

For example, if the communicative function of the fragment "C'est bon merci !" for a dimension called "social management" is "express gratitude", it cannot be any other in the same dimension, but it can be another in a different one, like "indicate problem solved" in the dimension "task management" for example.

2.2 Taxonomy

2.2.1 Dimensions

For this scheme, we define six semantic dimensions:

- Communication: dialogue acts about the communication process (e.g. *"J'ai fait une faute de frappe dans mon précédent mail c'est grep pas gret"*)

- Discourse: dialogue acts affect both the discourse's structure and the topics discussed (e.g. *"Avant de vous expliquer mon problème, j'ai un coup de gueule à faire passer"*)

- Social Obligations: dialogue acts that take care of social conventions such as thanks, greetings, apologies etc. (e.g. *"Merci beaucoup !!!!"*)

- Auto Feedback: dialogue acts related to the speaker's own processing of an utterance (e.g. *"Perso j'ai rien compris"*)

- Allo Feedback: dialogue acts related to the addressee(s)'(s) processing of an utterance (e.g. *"Je sais pas si je suis très clair...?"*)

- Task: dialogue acts that bring information about the task's objects, i.e. problems, solutions, goals, user profiles, contexts, etc. (e.g. *"Mon problème c'est que j'ai même pas l'icône dans le systray"*)

2.2.2 Communicative functions

Here we define the fourteen general communicative functions of the scheme. As was said before, they are all mutually exclusive. At most one per dimension can be attributed to the same functional segment. A segment can therefore bear up to six dialogue acts, however, in most cases, there is only one act per utterance, sometimes two, and rarely three or more.

These functions are sufficient to cover any utterance on any dimension.

Labels are written in capital letters, other items in the following bullet lists are merely function categories and subcategories, not to be used for annotation.

Information Transfer

Information transfer functions deal with the exchange of information between participants.

- Direct Information Transfer:
  These functions are neither backward nor forward looking, they provide information that was not directly elicited by a previous utterance.

  - $ASSERT$: if the speaker is the source of the information
    "Une nouvelle version d'Opera est sortie la semaine dernière."

  - $QUOTE$: if the speaker is not the source of the information
    "version : commande introuvable"

- Information Transfer Elicitation:
  These functions are forward-looking, they create an expectation for new information.

  - ASK: if the speaker expects another participant to provide the information
    "Comment ça se fait ?"

  - $SELF - ASK$: if the speaker is creating the expectation only to answer it himself (like a rhetorical question, for example)
    "Problème réglé ? Eh ben non !"

- Information Transfer Feedback:
  These functions are backward-looking, they provide information that relates to a previously uttered dialogue act.

  - $ANSWER$: if the information was directly elicited by a previous utterance that had a forward-looking information transfer func-

tion

"*C'est probablement parce que tu as oublié d'installer latex-mk*"

- – *REACT*: if the information was not directly elicited
  "*Ça me paraît pas très propre ta solution*"

Action planning

Action planning functions deal with the management of the participants' actions.

- Direct Action Planning
  These functions are neither backward nor forward looking, they either commit the speaker to an action or direct the addressee.

  - – *COMMIT*: commits the speaker to an action
    "*Je m'en charge*"

  - – *DIRECT*: directs the addressee to perform an action
    "*Signale le bug sur le tracker Mantis*"

- Action Planning Elicitation:
  These functions are forward-looking and elicit the addressees' approval of an action plan.

  - – *OFFER*: if the action is to be performed by the speaker
    "*Je peux revérifier si tu veux*"

  - – *REQUEST*: if the action is to be performed by the addressee
    "*Est ce que tu peux poster le contenu de ton fichier .bashrc?*"

- Action Planning Feedback:
  These functions are backward-looking and address an action planning elicitation function

  - – *ADDRESS REQUEST*: if the action is to be performed by the speaker
    "*Non c'est mort je n'ai même pas envie d'essayer*"

  - – *ADDRESS OFFER*: if the action is to be performed by the addressee
    "*Ah oui stp je veux bien que tu t'en charges*"

Other

- Performative Locution
  This function is not only describing a given reality but also actively changing it.

  - $DECLARE$: the utterance of a dialogue act having this function is, at least in part, an attempt at doing an action
    "J'abandonne"

- Self Expression
  This function expresses the speaker's psychological state, or his psychological position towards something.

  - $EXPRESS$:
    "Merci beaucoup !"

2.3 Extensions

While the six semantic dimensions and the fourteen communicative functions defined above are sufficient to cover virtually any conversation, more information may be necessary to effectively achieve more specific tasks, such as the modelization of problems and solutions in conversations bearing requests for assistance, or following the evolution of the mood a participant during his interactions.

2.3.1 Dialogue act dependencies

Our taxonomy of communicative functions includes several forward-looking and backward-looking functions that fall in either the information transfer or the action planning categories of functions. The former create an expectation (of information, of agreement, etc.) while the latter may satisfy these expectations. Being able to represent these expectations and to know whether they were satisfied or not would require the annotation of relational information about communicative functions. This is done by marking relationships between backward-looking functions and the utterance they relate to. By doing this, we can easily detect unsatisfied demands: there is one for each utterance bearing a forward-looking function that is not linked to a backward-looking dialogue act (or that is linked only to a partial backward-looking act).

The annotation of dialogue act dependencies in the Task dimension would be very useful in evaluating a problem-solving process.

2.3.2   Function qualifiers

Up to this point our annotation allow us to know what an utterance is *about* (the semantic content's dimension) and how it should be *processed* (the act's communicative function). However, additional data on utterances would be useful to reach an additional level of understanding of the dialogue.

[Petukhova and Bunt, 2010] introduces the notion of *qualifier*, that are used in conjunction with communicative functions to describe the utterance more precisely. They propose a representation of dialogue behaviour expressing intentions with different possible qualifications, relating to uncertainty, conditionality, partiality and mode.

The four qualifiers are:

- Modality: epistemic modal qualifiers specify the strength of the speaker's belief about the truth of a proposition (only information transfer functions are concerned by this qualifier)

- Conditionality: they represent the ability, necessity or volition of performing actions (only action planning functions are concerned by this qualifier)

- Partiality: they limits the scope of the communicative function to only a part of the dialogue act to which the current segment is related (only backward-looking functions are concerned by this qualifier)

- Mode: broad category of qualifiers concerned with the speaker's attitude and emotional state

For simplicity, we treat modality, conditionality and partiality qualifiers as binary values. Respectively, these values are: "certain" or "uncertain", "conditional" or "unconditional", and "partial" or "complete". There are several taxonomies available in the literature that could be used to label emotional and attitudinal phenomena in dialogue, however we choose to stay at a coarse level of granularity and use only "positive", "negative" and "neutral" as possible values for the mode qualifier.

To illustrate the use of qualifiers, let's take the following exchange as an example:

1. P1: Est ce que Unity et Gnome 3 sont disponibles sous Debian ?

2. P2: Il me semble que Unity n'est dispo que sous Ubuntu

3. P1: Bon si j'ai le temps ce week end je passerai sous Gnome alors

Using the previous qualifiers, it could be annotated as such:

1. Task: Ask

2. Task: Answer[partial, uncertain]

3. Task: Commit[conditional]

2.3.3 Problem and solution sub-objects

Problems and solutions are complex objects with a variety of properties. To be able to model them with increased accuracy, we may define the following sub-objects:

- Problem

    - $SYMPTOMS - DESCRIPTION$: description of the problem and how it affects the user and the system
      "Le software-center crash au lancement"

    - $GOAL - STATEMENT$: what the participant wants, but is prevented to achieve by the problem
      "J'aimerais bien utiliser Unity sous Debian"

    - $CONTEXT$: relevant information about the state of the world when the problem occurred
      "Je suis sous Ubuntu 12.04"

    - $CONTEXT - UPDATE$: relevant update concerning the state of the world
      "Maintenant j'arrive même plus à lancer le dashboard"

- Solution

    - $SOLUTION - INSTRUCTION$: the steps that are to be followed to solve the problem
      "Redémarre ton PC"

    - $SOLUTION - ENTITY$: the "thing" that needs to be acquired to solve the problem and that embodies the solution, may imply a solution-instruction
      "J'ai besoin d'écrire en dictant mon texte", "Utilise Dragon Natural Speaking"

- $SOLUTION - EXPLANATION$: the explanation of a problematic behaviour that implies a solution-instruction, may imply a solution-instruction
  "C'est parce que tu n'as pas lancé la commande en root"

- $SOLUTION - CONSTRAINT$: specifications on the solution by the participant(s) experiencing the problem
  "J'ai pas envie d'avoir à tout réinstaller"
- $SOLUTION - LIMITATION$: description of the solution's limits and shortcomings, as well as doubts about it's validity
  "Par contre c'est payant..."

The annotation of utterances of the Task dimension with these items, done in the same manner as if they were qualifiers, would be extremely useful. For example:

1. P1: Etant cloué au lit depuis un accident j'aurais aimé savoir si on peut utiliser la reconnaissance vocale sous linux ?

2. P1: Je suis sous Ubuntu 12.04

3. P1: Merci pour votre aide

4. P2: Il doit exister une licence professionnelle de Dragon Naturally Speaking

5. P3: Oui mais elle est sans doute très chère, et pas libre...

Could be annotated in the following manner:

1. Task: Assert{Goal-Statement}

2. Task: Assert{Context}

3. Social Obligations: Express

4. Task: Assert[uncertain]{Solution-Entity}

5. Task: React{Solution-Limitation}

With that information, two objects can be created and their different attributes "filled" with semantic information:

- Problem 1:
  - Goal: to use vocal recognition on linux
  - Context: a system operating Ubuntu 12.04

- Solution 1:
  - Solution-Entity: the software "Dragon Naturally Speaking"
  - Limitations: it's expensive, it's proprietary

2.3.4 Dimension-specific function specializations

In order to allow a more subtle definition of dialogue acts, their communicative function can be specialized into a dimension-specific label. These labels are all an extension of one of the fourteen general functions.

Assigning a function specialization is entirely optional, they are not necessary to annotate the text ; however they help in understanding how the semantic content of an utterance can be used. For example, it is useful to differentiate between a greeting and a farewell, even if both of the acts are declarations of the social obligations' dimension.

The following list is not exhaustive and subject to change.

- Assert
  - Social Obligations
    * $SELF-INTRODUCE$
      "Moi c'est Danny je viens de m'inscrire sur la liste"
  - Auto Feedback
    * $PROVIDE\ FEEDBACK$
      "J'ai rien compris !"
  - Allo Feedback
    * $PROVIDE\ FEEDBACK$
      "J'ai pas l'impression que tu aies bien compris"
  - Task
    * $REPORT$
      "J'ai essayé d'utiliser le software-center mais ça a crashé au milieu de l'installation"
    * $DESCRIBE$
      "Je suis sous Ubuntu 12.10"

* *CONSTRAINT*
  "Je préfererais ne pas avoir à réinstaller Ubuntu"
* *ENOUNCE LIMITATION(S)*
  "Par contre c'est payant"

- Quote

  - Task

    * *LINK RESOURCE*
      "https://help.ubuntu.com/lts/ubuntu-help/unity-launcher-intro.html"
    * *SHARE OUTPUT*
      "E: Impossible d'ouvrir le fichier verrou /var/lib/dpkg/lock"

- React

  - Task

    * *APPRAISE*
      "Ça ne marchera jamais"

- Ask

  - Auto Feedback

    * *ELICIT FEEDBACK*
      "Du coup il faut que je recommence tout, c'est bien ça ?"

  - Allo Feedback

    * *ELICIT FEEDBACK*
      "T'as compris ce que je voulais dire ?"

  - Task

    * *ELICIT APPRAISAL*
      "Je pense que ça devrait marcher, est ce que quelqu'un peut
      confirmer ?"

- Request

  - Task

    * *REQUEST HELP*
      "Donc si vous pouviez me donner des conseils ce serait sympa"

- Address Request

  - Task

* *ANNOUNCE TRIAL*
  "J'essaye dès que je rentre à la maison"

- **Declare**

  - Communication

    * *SWITCH CHANNEL*
      "Tant pis je vais demander sur le forum ce sera plus simple"

  - Discourse

    * *PRECLOSE*
      "Bon perso je vais pas tarder à y aller"
    * *OPEN*
      "bip Popaul j'ai une question si t'es toujours là"
    * *INTRODUCE TOPIC*
      "J'aimerais bien qu'on se refocalise sur Gnome 3 perso"
    * *SHIFT TOPIC*
      "Pour en revenir à Unity..."

  - Social Obligations

    * *GREET*
      "Bonjour tout le monde"
    * *FAREWELL*
      "A+"

  - Task

    * *FORFEIT*
      "J'abandonne"
    * *UPDATE STATUS*
      "Ok c'est bon ça marche !"

- **Express**

  - Social Obligations

    * *EMOTE*
      ";-)"
    * *WISH*
      "Bonne soirée !"
    * *DOWNPLAY*
      "Mais de rien"

* *THANK*
  "Merci beaucoup !"
* *APOLOGIZE*
  "Vraiment désolé..."

Appendices

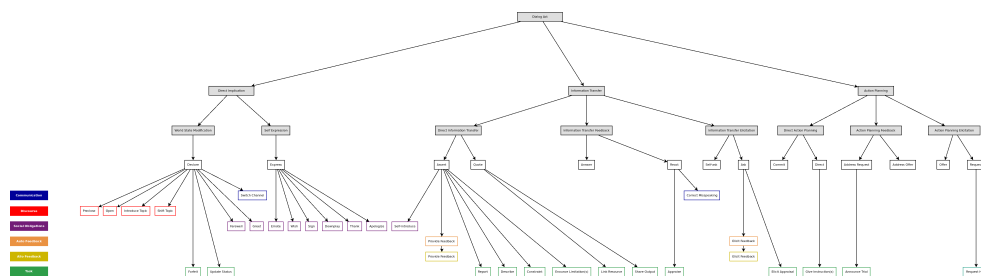Graphical representation of the taxonomy (attached file: *taxonomy.png*)

Figure 2: Graphical taxonomy preview

References

[Bunt, 2006] Bunt, H. (2006). Dimensions in dialogue act annotation. In *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*.

[Bunt, 2009] Bunt, H. (2009). The dit++ taxonomy for functional dialogue markup. In *AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts*, pages 13–24.

[Bunt, 2011] Bunt, H. (2011). The semantics of dialogue acts. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 1–13. Association for Computational Linguistics.

[Hernandez and Salim, 2014] Hernandez, N. and Salim, S. (2014). Exploiting the human computational effort dedicated to message reply formatting for training discursive email segmenters. *LAW VIII*, page 110.

[Petukhova and Bunt, 2010] Petukhova, V. and Bunt, H. (2010). Introducing communicative function qualifiers.

[Traum and Larsson, 2003] Traum, D. R. and Larsson, S. (2003). The information state approach to dialogue management. In *Current and new directions in discourse and dialogue*, pages 325–353. Springer.