Customer Interaction 2.0: Adopting Social Media as Customer Service Channel

Michaela Geierhos
Center for Information and Language Processing, University of Munich, Germany
Email: michaela.geierhos@cis.uni-muenchen.de

***Abstract*—Since customers first share their problems with a social networking community before directly addressing a company, social networking sites such as Facebook, Twitter, MySpace or Foursquare will be the interface between customer and company. For this reason, it is assumed that social networks will evolve into a common communication channel – not only between individuals but also between customers and companies. However, social networking has not yet been integrated into customer interaction management (CIM) tools. In general, a CIM application is used by the agents in a contact centre while communicating with the customers. Such systems handle communication across multiple different channels, such as e-mail, telephone, Instant Messaging, letter etc. What we do now is to integrate social networking into CIM applications by adding another communication channel. This allows the company to follow general trends in customer opinions on the Internet, but also record two-sided communication for customer service management and the company's response will be delivered through the customer's preferred social networking site.**

***Index Terms*—social media business integration, multi-channel customer interaction management, contact centre application support**

I. INTRODUCTION

With the introduction of social networks such as Facebook and Twitter customer communication habits are changing. Considering that Facebook alone has more than 672 million users in April 2011 [1] it becomes apparent that Facebook currently is the most preferred medium by consumers and companies alike. Since many businesses are moving to online communities as a means of communicating directly with their customers, social media has to be explored as an additional communication channel between individuals and companies.

Using the traditional communication channels such as telephone and e-mail, there are already established approaches and systems to incoming requests. They are used by companies to manage all client contacts through a variety of mediums such as telephone, fax, letter, e-mail and online live chat. Contact centre agents are therefore responsible to assign all customer requests to internal business processes (e.g. technical service, shipping, marketing, and accounting). This process is very time-consuming and error-prone. It can therefore happen that customer requests "stray" for a long time in the company before reaching the person responsible for it, which can sometimes lead to loss of clients [2]. But with the introduction of Web 2.0 and social networking customers are more likely to communicate with the companies via Facebook and Twitter instead of filling data in contact forms or sending e-mail requests.

Almost all large enterprises, especially in the mobile phone sector, already have regularly updated Facebook pages for customer service and support. Until now, contact centre agents still read these wall postings and forward them to the persons in charge of customer services in the company. Even though hundreds of social media monitoring tools exist (e.g. GoogleAlerts by Google, BuzzStream by BuzzStream, Sysomos by Sysomos Inc., Alterian by Alterian, Visible Technologies by Visible Technologies, and Radian6 by Radian6 Technologies Inc.), there is still no systematic, technically integrated solution that provides full support starting with social media monitoring through to direct company response on the Facebook wall. We therefore developed a modular system that is completely web based and combines all technologies, data, software agents and human agents involved in the monitoring and customer interaction process. It therefore provides maximum flexibility to service providers by enabling multiple services with only one system.

Our proposed solution towards a web monitoring and customer interaction management system is quite simple. We focus on a modular architecture fully configurable for all components integrated in its work-flow (cf. Fig. 1 and Fig. 2). Although our first prototype is designed for processing customer messages posted on social networking sites about mobile phone specific issues, it can also deal with other topics and use different text types such as e-mails, blogs, RSS feeds etc. Unlike the commercial monitoring systems mentioned above, we concentrate on a linguistic, rule-based approach for message classification and product name recognition. One of its core innovations is its paraphrasing module for intra-and inter-lingual product name variations because of different national and international spelling rules or habits. By mapping product name variations to an international canonical form, our system allows for answering questions like *Which statements are made about this mobile phone in which languages/in which social networks/in which countries/…?* Moreover, our system can be adapted according to user's language needs, i.e. the application can be easily extended on further natural

This paper is based on "SCM – A Simple, Modular and Flexible Customer Interaction Management System" by J. Schuster, Y. S. Lee, D. Kobothanassi, M. Bargel, and M. Geierhos, which appeared in the Proceedings of the International Conference on Information Society (i-Society 2011), pp. 169-174, June 27-29, 2011, London. ISBN: 978-0-9564263-8-3. Copyright © i-Society 2011 Technical Co-Sponsored by IEEE UK/RI Computer Chapter.

doi:10.4304/jait.2.4.222-233

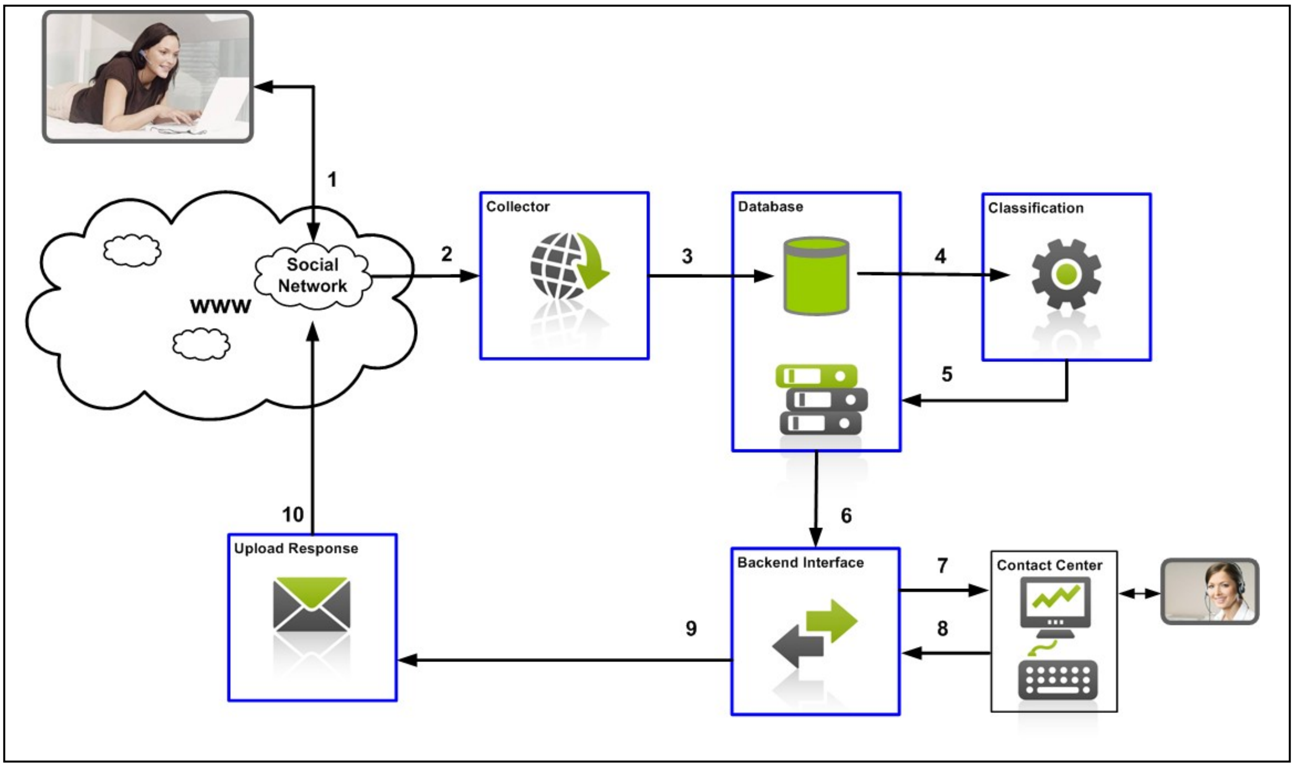


Figure 1. **System architecture.** A preselected Social Network will be the interface between customer and company. (1) For instance, Facebook users post on the wall of a mobile phone company messages concerning tariffs, technical malfunction or bugs of its products, positive and negative feedback. (2) The collector should download every n seconds (e.g. 10 sec) data from the monitored social networking site. Above all it should be possible to choose the social networking site, especially the business pages, to be monitored. This can be configured by updating the collector's settings. In order to retrieve data from Facebook, we use its graph API. (3) The customer messages will be stored in a database. (4) After simplifying their structure, the requests have to be categorized by the classification module. (5) During the classification process, we assign both content and semantic tags as features to the user posts before re-storing them in a database. (6) According to the tags the messages are assigned to the corresponding business process. (7) This n:1 relationship is modelled in the contact centre interface before passing these messages as e-mail requests to the customer interaction management tool used in contact centres. Finally, the pre-classified e-mails are automatically forwarded to the persons in charge of customer services. (8) Those agents reply to the client requests and their responses will be delivered via e-mail to the contact centre before being transformed to social network messages and (9) send back to the Facebook wall. (10) Afterwards, the Facebook user can read his answer.

languages. Until now, our prototype can deal with three very different languages: German, Greek and Korean.

The paper is structured as follows. In Section II we discuss some related work. In Section III we give an overview of our system, define its constraints and go into its configuration. Using some examples of German we show the phenomenon of linguistic variation in social communication in Section IV. How important linguistic processing of customer requests is, will be shown in Section V when explaining how our named entity paraphrasing engine works. Section VI is about training data and how the content and sentiment classifiers are developed and applied. Finally, we suggest future directions for our research.

II. Related Work

SCM (***S**ocialCom **C**ustomer **I**nteraction **M**anagement*) is a system that was created in a collaborative project (called *SocialCom*) between research and industry. We therefore consider two types of related work. Previous work can, on one hand, be found in scientific articles about text classification and named entity normalization. On the other hand, the systems of the various companies that offer customer interaction management and web monitoring must be also considered.

The current trend is to develop virtual contact centers that integrate a company's fan profiles into social networking sites. The virtual contact centers process the customer contacts and forward them to the company's service and support team. Eptica by EADS e.g. provides a commercial tool for customer interaction management via Facebook. Other systems try to predict election results [3] or success of movies and music [4] by using scientific analysis of opinion polls or by doing sentiment analysis on special web blogs or online forum discussions. Further relevant issues are topic and theme identification and sentiment detection. Since blogs consist of news or messages dealing with various topics, blog content has to be divided into several topic clusters. Pal and Saha [5] from Hewlett Packard developed the "Best Separators Algorithm" which cuts documents into blocks according to theme relevance.

As scientific articles are concerned, there are basically two topics that are relevant to our work: Text classification and named entity normalization. There does not seem to be much literature about the classification of texts of the type we dealt with. An interesting description of a blog classification system (blogs are much closer in many respects to the texts we classified than most other text types) is given in [6]. Gilad Mishne developed "a method for tagging blog posts based on a collaborative filtering approach: given a blog post, tags for it are

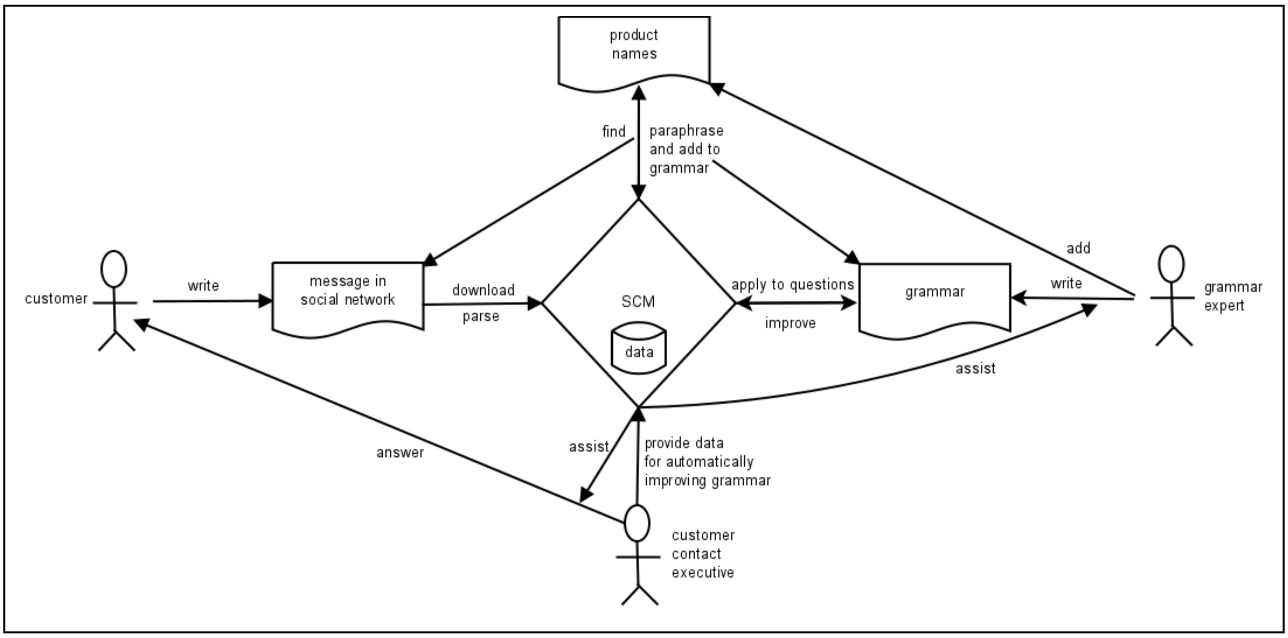


Figure 2. **System design.** Customers post messages on a social networking site. These messages typically concern products and/or services sold or provided by a company. Most messages state that a problem of a certain type occurred with a certain product or service. Some messages also contain expressions of praise or thank. SCM downloads all messages and stores them in a normalized format. It applies its grammar to the messages in order to recognize all product names in it and in order to assign a set of content tags (such as *hotline*, or *delivery* or *software available?*) and a sentiment tag (at the moment either *positive* or *negative*). The message, their tags and the list of product names are then forwarded to those customer contact executives that are experts for the recognized type of problem. If a message has been assigned a wrong tag or no tag at all, customer contact executives can assign a tag manually. They have to mark those parts of the message that justify its being assigned the tag. Thus, customer contact executives unknowingly add new constraints to the grammar.

suggested according to tags assigned to other, similar posts" [6, p. 101]. As the tags that our texts are supposed to receive usually strongly depend on "named entities" (such as product or tariff names) and other strings that occur in them, we used a different approach.

As concerns proper name or named entity normalization, two articles are especially interesting: Jijkoun et al. discuss a method for "named entity normalization in user generated content" in [7]. After giving an overview of the literature about named entity normalization [7, p. 24], they discuss "five improvements to [a] baseline NEN [i.e. named entity normalization] algorithm" [7, p. 23]. One of their ideas is to use anchor texts in Wikipedia articles to find variants of named entities. In [8], Hema Raghavan and James Allan compare seven (mostly statistical) methods for normalizing proper name variants. For two reasons, we chose to use an approach that differs from Raghavan's and Allan's techniques as well as from the approach used by Jijkoun et al.: First the variants of product and tariff names usually follow rather obvious and regular patterns. We felt it would be unnecessarily complicated to use statistical methods for them. Secondly, the variants of the names of mobile phone related products and tariffs cannot be found in Wikipedia.

III. System Overview

SCM is a web based monitoring and customer interaction management system that combines all technologies, data, software agents and human agents involved in the monitoring and customer interaction process, such as the technologies used for downloading, processing and storing data, the software agents used for classification and similar tasks and the human agents such as grammar experts and customer contact executives.

A. Non-Functional Requirements

Using the following trigger questions, we define a list of constraints being valid for our system.

Human Factors:

Q *Who will be using the system?*

Customer contact executives (cf. Fig. 2), especially contact centre agents (cf. Fig. 1), will use this application.

Q *Will more than one type of user be using the system?*

On the one hand, persons in charge of customer services will be the main users of our system. But we should not forget the grammar experts (cf. Fig. 2) on the other hand. They are responsible for the system configuration, add new product names and improve the grammar setting for classification purposes.

Q *What kind of training will be required for each type of user?*

No additional training will be necessary because contact centre agents are already used to e-mail communication. Besides, grammar experts are highly trained specialists and we will need their advanced technical know-how only from time to time.

Q *Are users protected from making mistakes?*

It is not foreseen that contact centre agents perform the classification step on their own. They can only

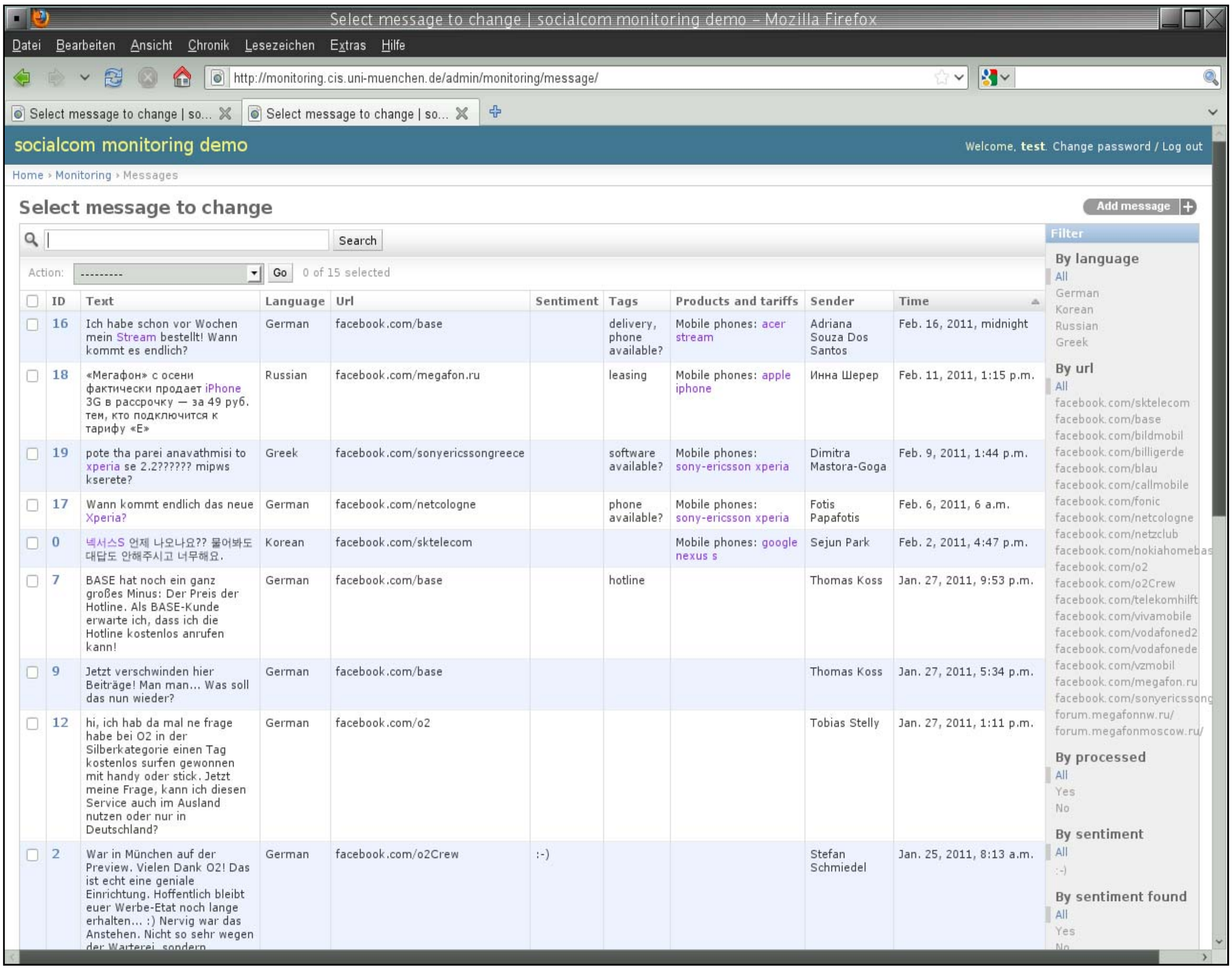


Figure 3. Management of multilingual customer requests

react on pre-classified messages in their inbox and forward them in case of incorrect assignment to the person in charge. This reduces the chance of making mistakes or unreasonable settings.

User Interface:

Q *Why is it important to create a user friendly interface?*

An intuitive user interface is absolutely necessary. As shown in Fig. 3 the contact centre agents get pre-classified consumer requests (cf. *tags*, e.g. *"delivery, phone available?"*, *"leasing"*, *"software available"*, or *"hotline"*) from a monitored social networking site (e.g. Facebook). Our tool provides additional information such as the *sender*'s name, the *product name* mentioned in the *text* (e.g. *"iPhone"*), a normalized form of the *product and tariff names* (e.g. *"apple iphone"*), the *sentiment* of the customer post (positive ':-)' or negative ':-('), the *URL* from where the message was retrieved and the *time* when it was posted. Furthermore, the *language* of the request is identified (e.g. German, Russian, Greek, and Korean).

System Interfacing:

Q *Is input coming from systems outside SCM?*

Since our tool has established a connection with the monitored social networking site, consumer requests are coming as input stream into the system. Afterwards the classification and assignment steps will performed by the application itself.

Q *Is output going to systems outside SCM?*

Our approach emulates an issue tracking system delivering e-mails with Facebook posts to the corresponding person in charge of customer services. We therefore have to plug our system into a support ticket system.

Q *Are there restrictions on the standard input format?*

For example, Facebook wall posts are represented as structured data that can easily be retrieved from Facebook graph API, for example, from https://graph.facebook.com/telekomhilft/feed. We simplify this data format before using it for extraction and classification purposes.

System Modifications:

Q *What parts of the system are likely candidates for later modification?*

If the Facebook graph API changes, we will have to update the module collecting the customer requests.

Moreover, the product types and names can change in the telecommunication sector. We therefore have to retrain the classificator from time to time.

Q *What sorts of modifications are expected?*

We expect modifications in configuration. Upgrades will be necessary with respect to fluctuating (domain-dependent) vocabulary used for classification.

Performance Characteristics:

Q *Are there any time constraints on the system?*

Every *n* seconds consumer posts should be retrieved from the monitored social networking site.

Error Handling:

Q *How should the system respond to input errors?*

As input error we consider the website unavailability of the monitored social network. In this case, our application has to re-establish the connection to Facebook and retry to retrieve consumer posts. Furthermore, the contact centre agents will get status updates of the website availability.

These system design issues are essential for the architecture of the application. We only discussed the most important questions by skipping non-functional requirements such as recoverability, security, backup, capacity, documentation and all the rest of it.

B. Language Selection: Why not English?

SCM can be used with any natural language. In our prototype instance of SCM, we created classifiers and product names for German, Korean and Greek. *But why?*

While the English speaking consumers on Facebook are more likely to respond to communication rather than to initiate communication with an organisation [9], the German speaking community in turn directly contacts the companies. Therefore, some German companies – especially in the telecommunication sector already have regularly updated Facebook pages for customer service and support, e.g. Telekom (German Telecom), Vodafone, O2 or Nokia.

Figure 4. Facebook wall for customer service and support provided by German Telecom.

Since the German and, for example, the Australian Facebook communities are complete opposites behaviourally because 89% of the Australian consumers only respond to communication and only 11% initiate communication with a company [9], we cannot concentrate on the English speaking Facebook users. Due to our observations of customers' behaviour on Facebook in the US and the UK, we must admit that the Australian pilot study [9] is quite representative for the English speaking Facebook communities. We therefore focus on German Facebook users searching the dialogue with companies' customer support services.

In order to prove that our system can be used for messages written in any natural language, we also selected Korean for our prototype because it is written in Hangeul (the Korean alphabet) and Greek because it is written with both alphabets: the Greek and the Latin alphabet.

C. Configuration

Since customers first share their problems with a social networking community before directly addressing the company, the social networking site will be the interface (cf. Fig. 1) between customer and company. For instance, Facebook users post on the wall of a mobile phone company (cf. public Facebook wall of German Telecom in Fig. 4) messages concerning tariffs, technical malfunction or bugs of its products, positive and negative feedback. In order to retrieve data from Facebook, we use its graph API as described above.

Actually, SCM does not only contain one grammar as shown in Fig. 2. It contains as many grammars as there are classifier objects. Each classifier has a (content or sentiment) tag, a set of positive and a set of negative constraints. (Fig. 10 shows some simple content classifiers and the grammars.) If one of the positive constraints holds, the classifier's tag is assigned. If one of the negative constraints holds, the classifier's tag is not assigned, irrespective of the positive constraints. I.e. negative constraints can be used to override positive constraints.

Put simply, the constraints are regular expressions. Yet, as they can contain a virtually unlimited number of special terms that refer to other grammars (such as `__mobile_phone__`), it would be more adequate to call them Local Grammars in the sense of [10]–[12]. Each instance of SCM comes with a predefined set of such special terms. (The idea is to tailor new instances of SCM to the needs of the companies or organizations that use it.) Grammar experts cannot add special terms or change their meaning. They just use them: If a grammar line, for example, contains `foo __mobile_phone__ bar`, it will match `foo`, followed by a space character, followed by any variant of any mobile phone name known to the used instance of SCM, followed by a space, followed by `bar`. Fig. 2 gave an overview of SCM's design. Its core technologies, the product name paraphrasing engine and the language-specific grammars for classification purposes, are described in more detail in later sections.

IV. LINGUISTIC PROCESSING OF GERMAN CUSTOMER REQUESTS ON FACEBOOK

Before thinking about the classification approach we have to study the linguistic phenomena typically appearing in social communication. But there is one basic prerequisite we have to discuss first: the notion of *social network sublanguage*.

Sublanguages are specialised language subsets which are distinguished by the special vocabulary and grammar from the general language [13], [14]. With respect to social networks, a sublanguage is characterised by a certain number of phrases or a grammar and special vocabulary [15], e.g. *"Gingerbread-Handy"*.

A. Sublanguage analysis

Social network sublanguages occur on the Facebook walls as well as on Twitter and other social networking sites. Regarding the Facebook pages we try to discover relationships between clients and products, customers and technical problems, products and features that will be used for classification purposes.

But the variety of organization-specific standard phrases and vocabulary that frequently emerge on Facebook walls are clustered into *attribute classes* during the training step of our system. For instance, the classes *"manufacturer"* (cf. Fig. 5), *"brand name"*, *"mobile phone model"* (cf. Fig. 6) and *"mobile accessories"* totally contain about 4,000 specialised words and phrases.

```
ericsson,.N+HERSTELLER
escada,.N+HERSTELLER
exopc,.N+HERSTELLER
FIC,.N+HERSTELLER
fifth media sdn. bhd.,fifth media.N+HERSTELLER
fifth media,.N+HERSTELLER
first international computer,FIC.N+HERSTELLER
fitage gmbh & co. kg,fitage.N+HERSTELLER
fitage,.N+HERSTELLER
freenet,.N+HERSTELLER
fujitsu siemens,.N+HERSTELLER
fujitsu technology solutions,fujitsu siemens.N+HERSTELLER
fujitsu,fujitsu siemens.N+HERSTELLER
g.b.t. technology trading,gigabyte.N+HERSTELLER
garmin deutschland,garmin.N+HERSTELLER
garmin,.N+HERSTELLER
garmin-asus,.N+HERSTELLER
GE,general electric,.N+HERSTELLER
general electric,.N+HERSTELLER
general mobile,.N+HERSTELLER
giga-byte,gigabyte.N+HERSTELLER
goldvish,.N+HERSTELLER
google,.N+HERSTELLER
grundig intermedia,grundig mobile.N+HERSTELLER
grundig mobile,.N+HERSTELLER
grundig,grundig mobile.N+HERSTELLER
```

Figure 5. Sample instances of the attribute class *"manufacturer"* providing part-of-speech (N) and semantic (HERSTELLER) information. (HERSTELLER is German for *"manufacturer"*.)

There are more linguistic phenomena apart from domain-specific vocabulary we have to deal with when analysing social network communication such as orthographic variation, misspelling, creative grammar application, neologism (e.g. *"SW flashung"* instead of *"SW flashing"*), various product names (e.g. *"Sony Ericsson Xperia Neo"*, *"Nokia Extrapower DC-11"*, *"HTC Desire"*) and even capitalisation is ignored.

All attribute classes (cf. sample classes in Fig. 5 and 6) form closed semantic classes characterised by lexica of limited-size. We created several specialised dictionaries for simple terms as well as for multi-word terms which can deal with a substantial part of the above mentioned sublanguage and which conform to the DELA lexicon format [16], [17] and which are modelled as finite automata [18].

```
a1000,motorola a1000.N+HANDY_MODELL+MOTOROLA
motorola a1000,.N+HANDY_MODELL+MOTOROLA
a1010,motorola a1010.N+HANDY_MODELL+MOTOROLA
motorola a1010,.N+HANDY_MODELL+MOTOROLA
a1200,motorola a1200.N+HANDY_MODELL+MOTOROLA
motorola a1200,.N+HANDY_MODELL+MOTOROLA
a1200 ming,motorola a1200 ming.N+HANDY_MODELL+MOTOROLA
motorola a1200 ming,.N+HANDY_MODELL+MOTOROLA
a1600,motorola a1600.N+HANDY_MODELL+MOTOROLA
motorola a1600,.N+HANDY_MODELL+MOTOROLA
a1800,motorola a1800.N+HANDY_MODELL+MOTOROLA
motorola a1800,.N+HANDY_MODELL+MOTOROLA
a388c,motorola a388c.N+HANDY_MODELL+MOTOROLA
motorola a388c,.N+HANDY_MODELL+MOTOROLA
a630,motorola a630.N+HANDY_MODELL+MOTOROLA
motorola a630,.N+HANDY_MODELL+MOTOROLA
a668,motorola a668.N+HANDY_MODELL+MOTOROLA
motorola a668,.N+HANDY_MODELL+MOTOROLA
a688,motorola a688.N+HANDY_MODELL+MOTOROLA
motorola a688,.N+HANDY_MODELL+MOTOROLA
a728,motorola a728.N+HANDY_MODELL+MOTOROLA
```

Figure 6. Sample instances of the attribute class *"mobile phone model"* providing part-of-speech (N) and semantic (HANDY_MODELL+MOTOROLA) information. (HANDY MODELL is German for *"mobile phone model"*.

The convention of dictionary development according to the DELA format enables the use of Local Grammars [11], [12] within the LGPL (GNU Lesser General Public License) software Unitex [19]. This platform provides all linguistic tools necessary for the processing of big corpora and enables the efficient handling of electronic lexica. Additionally, the development of Local Grammars, represented by graph structures (cf. Fig. 7), is supported by a graphical development tool.

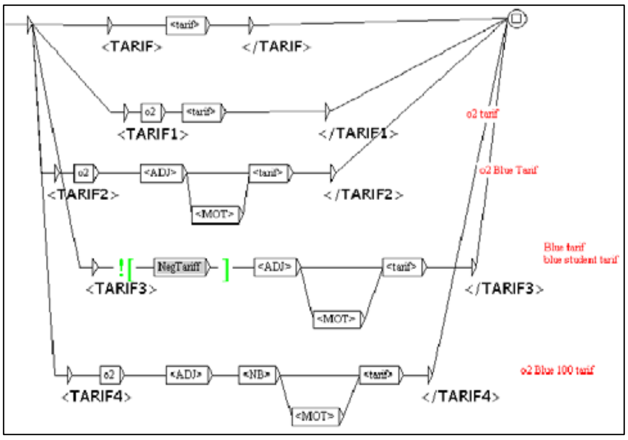


Figure 7. **Local Grammar for mobile phone tariffs.** This finite state transducer shows a regular expression used for the recognition of German mobile phone tariffs where <NB> stands for a numeric group, <MOT> for an alphabetic group and <ADJ> for an adjective listed in lexicon for German. We use negative constraints to exclude ambiguous tariff names: "NegTariff" is a Local Grammar included in this one.

B. Sublanguage and Local Grammars

Hunston and Sinclair [20] showed that it is possible to consider Local Grammars as small sublanguages and therefore, for a given domain (here, telecommunication in social networks), it is possible to build large-scale Local Grammars covering at best the entire sublanguage.

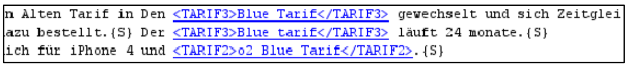

n Alten Tarif in Den <TARIF3>Blue Tarif</TARIF3> gewechselt und sich Zeitglei
azu bestellt.{S} Der <TARIF3>Blue tarif</TARIF3> läuft 24 monate.{S}
ich für iPhone 4 und <TARIF2>o2 Blue Tarif</TARIF2>.{S}

Figure 8. Mobile phone tariffs recognised by the Local Grammar shown in Fig. 7

Formally, Local Grammars are recursive transition networks [21]. The construction and manipulation of those are facilitated by the free software tool Unitex [19]. They are not intended to describe the entire grammar of a language, but they can be successfully used to describe the syntax and vocabulary of linguistic phenomena related to a specialised domain [22]. In this context they perfectly fit into the extraction process and enable us to restrict our analysis to the relevant facts.

From a technical point of view, Local Grammars are used in a cascading style. Each level of the cascade relies on the results of the previous level. The first levels enable us to identify and annotate simple entities (e.g. person names, products, tariff).

C. Internal and external indicators for NER

"Internal evidence is derived from within the sequence of words that comprise the name. (...) By contrast, external evidence is the classificatory criteria provided by the context in which a name appears" [23].

Others experimented with several lexicon sizes and discovered that a large comprehensive lexicon cannot improve considerably the precision or recall of a NER system [24]. Hence, we also pursue this strategy and compile the internal and external indicators into the corresponding attribute classes. Moreover, the list of indicators is open-ended and managed within different files – a sub-list per attribute class (e.g. *"manufacturer"* (cf. Fig. 5), *"brand name"*, *"mobile phone model"* (cf. Fig. 6) and *"mobile accessories"*. Some examples for external indicators are shown below:

TABLE I.
EXAMPLES OF EXTERNAL INDICATORS FOR ATTRIBUTE CLASSES

Attribute Class	Vocabulary
manufacturer	XY *Handy* (XY mobile phone; e.g. `Nokia`) XY *Vertragshandy* (XY contract phone; e.g. `Motorola`) XY-*Vertrag läuft aus* (XY contract coming to an end; e.g. `Apple`)
mobile phone	*Vertragshandy* XY *defekt* (Contract phone XY out of order) *verschicke MMS mit meinem* XY (send MMS via XY; e.g. `Apple iPhone`) *kaufe mir ein* XY (*buy an* XY; e.g. `Samsung Galaxy`)

V. INTERNATIONAL PRODUCT NAME PARAPHRASING

Within the customer messages, we try to discover relationships between clients and products, customers and technical problems, products and features that will be used for classification purposes. We are aware of the fact that many products (mobile phones, chargers, headsets, batteries and software) are sold in different countries under the same or under different names. Our system stores a unique international ID for each product. Product names and their paraphrases are language specific. Our prototype normalizes found product names to the international ID.

A. Paraphrasing classes

Our first approach to product name paraphrasing was to use paraphrasing classes. Much as verbs are inflected according to their inflection class, product names were inflected according to their paraphrasing class. Yet, paraphrasing classes had to be assigned manually and quite many classes were needed. Therefore, we decided to use a simplified system: Each product or manufacturer name is stored in a canonical form: Thus, a name of the type *glofiish g500* is stored in the form *glofiish-g-500*, even if *glofiish g-500* or *glofiish g500* should be more frequent. The minus characters tell our system where a new part of the product name begins. A product or manufacturer name has *permutations*: In German *o2 online tarif* has the permutation *tarif o2 online*. Standard permutations are added automatically: A product or manufacturer name with three parts has the standard permutation *123*. German tariff names of the type *o2 online tariff* have the standard permutations *312* and *23 von 1* as in *online tarif von o2* (*online tariff by o2*). Grammar experts can add more permutations manually.

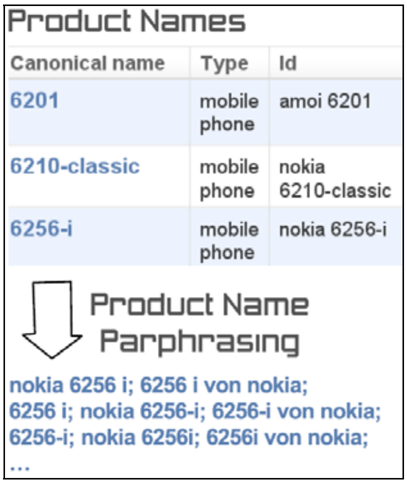
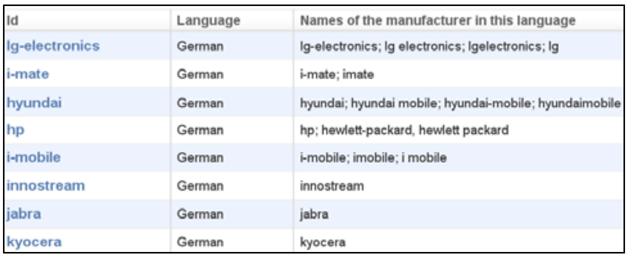


Figure 9. German product name paraphrasing for *"nokia 6256-i"*



Id	Language	Names of the manufacturer in this language
lg-electronics	German	lg-electronics; lg electronics; lgelectronics; lg
i-mate	German	i-mate; imate
hyundai	German	hyundai; hyundai mobile; hyundai-mobile; hyundaimobile
hp	German	hp; hewlett-packard, hewlett packard
i-mobile	German	i-mobile; imobile; i mobile
innostream	German	innostream
jabra	German	jabra
kyocera	German	kyocera

Figure 10. German paraphrasing classes for manufacturers (excerpt)

B. Spelling variants

Apart from their canonical name and its variants, product names can also have spelling variants. Thus, *android* has the spelling variants *androit*, *andoid* and *andoit*. (These are some of the most frequent ways *android* is actually spelt in the customer messages.) For each spelling variant, our system automatically generates all paraphrases that exist according to the standard and the manually added permutations of the canonical name. I.e. the paraphrases of the mobile phone name *e-ten glofiish-g-500* include *e-ten klofisch-g-500, e-ten klofisch-g 500, e-ten klofisch g-500*, etc.

C. Lexical variants

Apart from spelling variants, product names can also have lexical variants. The mobile phone *tct-mobile-ot-v-770-a* has the lexical variant *playboy phone*. The regular permutation transformations are not applied to lexical variants. But lexical variants and their manufacturer-based variants (e.g. *tct playboy phone* and *playboy phone*) are, of course, paraphrases, too.

D. Replacements

Depending on which characters are defined to be possible replacements of the minus character (the empty character, space, comma, etc.), SCM will generate a greater or smaller number of strings that actually never occur in the customer messages. But, for several reasons, over-generation is not a problem. First, grammar experts can manually exclude variants. Secondly, many of the variants that never occur in the messages (such as *tct.mobile-ot,v770.a*) will hardly be used with the intent to refer to something else than the product of whose name they are claimed to be a paraphrase. Besides, product names are only forwarded to human agents if they have been found in a message that has been classified as a message of a certain content type. I.e. product names are only recognized in contexts that are typical for the type of product name in question.

E. Product name identification

SCM's product name paraphrasing engine is aware of the fact that many products are sold in different countries under the same or under different names. Our system stores a unique international ID for each product. Product names and their paraphrases are language specific. SCM normalizes found product names to the international ID. This way, it provides the data for answering questions such as *"Which statements are made about this mobile phone in which languages/in which social networks/in which countries/…?"*

VI. GRAMMAR-BASED CLASSIFIACTION

Since we analyse Facebook wall posts in the telecommunication sector, we started to create a corpus of consumer requests by collecting messages around the topic *"mobile phone"*.

A. Training data

In the absence of a reference corpus or any benchmark, we build a corpus containing about 73,000 words. These texts were gathered via the Facebook graph API, reduced to the customer messages and later manually labelled.

B. Applying local grammars for feature extraction

We used linguistically motivated local grammars to describe telecommunication contexts of customer requests and to detect the product and tariff identifiers within these contexts. Local grammars [11], [12] enable the description of a local context and restrict the emergence of certain lexical or syntactical features to a window of predefined size. Thus, they avoid or reduce ambiguities that occur for a simple keyword search.

In order to apply these graphs (e.g. Fig. 7 amongst others), the clients' Facebook wall posts have to be passed through the following processing pipeline of the Unitex system [19]:

1) *Conversion*: Converts the text into Unicode (UTF-16-LE)
2) *Normalization*: Normalizes the special characters, white spaces and line breaks
3) *Sentence*: End of sentence detection
4) *Tokenization*: Tokenizes the text according to the alphabet of the investigated language
5) *Dictionary*: Performs a lexical analysis by comparing tokens to lexical entries and assigns each word to its possible grammatical categories
6) *Concordance*: Construct concordance or merge transducer output with original text

Unitex takes as input plain text in UTF-16-LE and gives as output a concordance or a merged text. Technically speaking there is also detailed information available on the matches and their offsets. This information is usually used by the Unitex module *Concord* [19] but can be used in one's own program as well.

The lexical analysis is directly followed by the step of information extraction via local grammars. Each piece of structured information is covered by different extraction units such as person, product, tariff, mobile phone model, and manufacturer. We created a system of local grammars which works iteratively and sometimes cascaded [25]. The cascaded grammars are executed with different priorities. For example, we developed local grammars to extract job descriptions with eight levels of priority. The graphs of level $n + 1$ are only applied if the grammars of level n with higher priority generate no recall.

The main task of feature extraction is to normalize the extracted sequences and to eliminate duplicates as well as incompletely recognised units. Because sequences are recognised simultaneously by alternative paths of the graphs that work as annotation transducers a decision on the match strategy had to be made. Since results on training data showed a superior performance, the longest match strategy was used.

C. Grammar Development and Application

Grammar experts can create any number of content and sentiment classifiers. As already pointed out, a classifier's grammar consists of a set of positive constraints and a set of negative constraints. To classify a message, SCM simply applies the grammars of all its

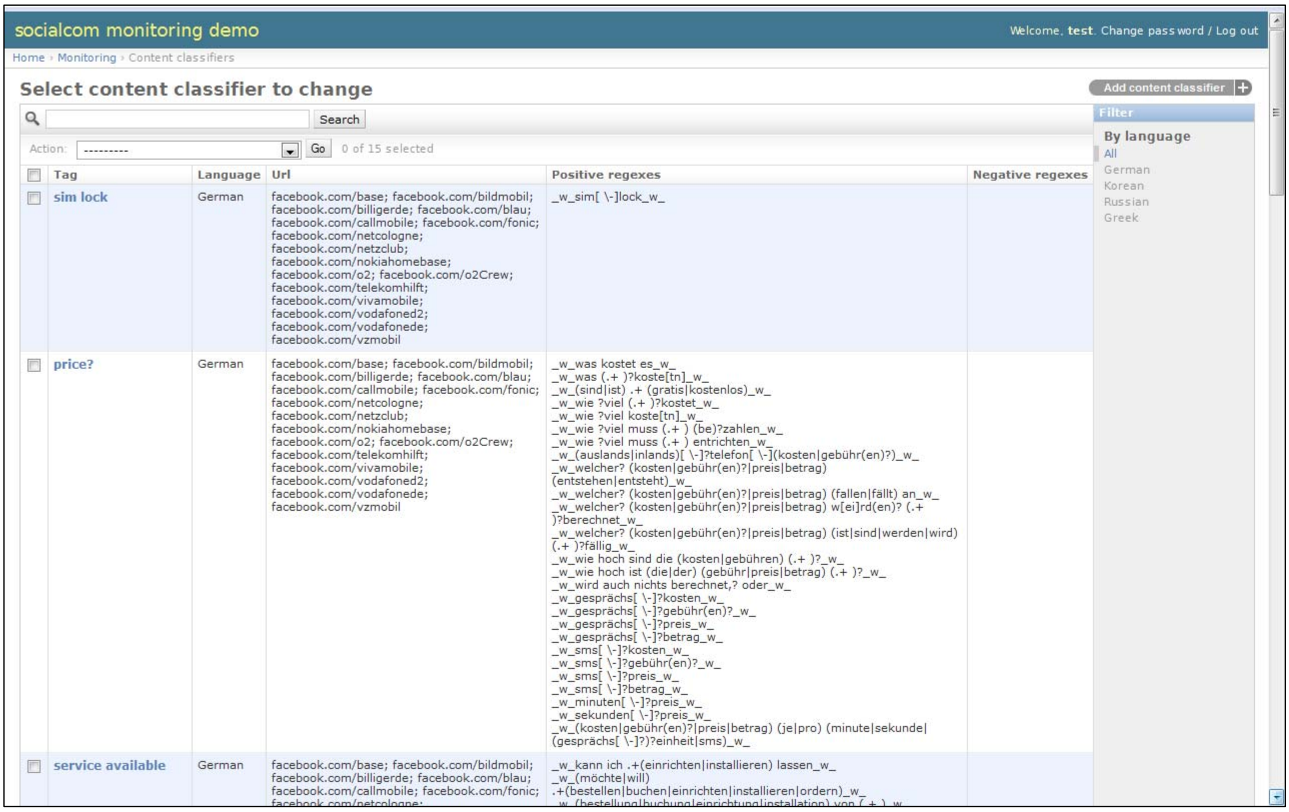


Figure 11. Content classifiers in an SCM instance

classifier objects to the message. If a content classifier's grammar matches, its tag is added to the message's content tags. Sentiment classification works analogously with the exception that exactly one tag is assigned. Content and sentiment classificators are language and URL specific:

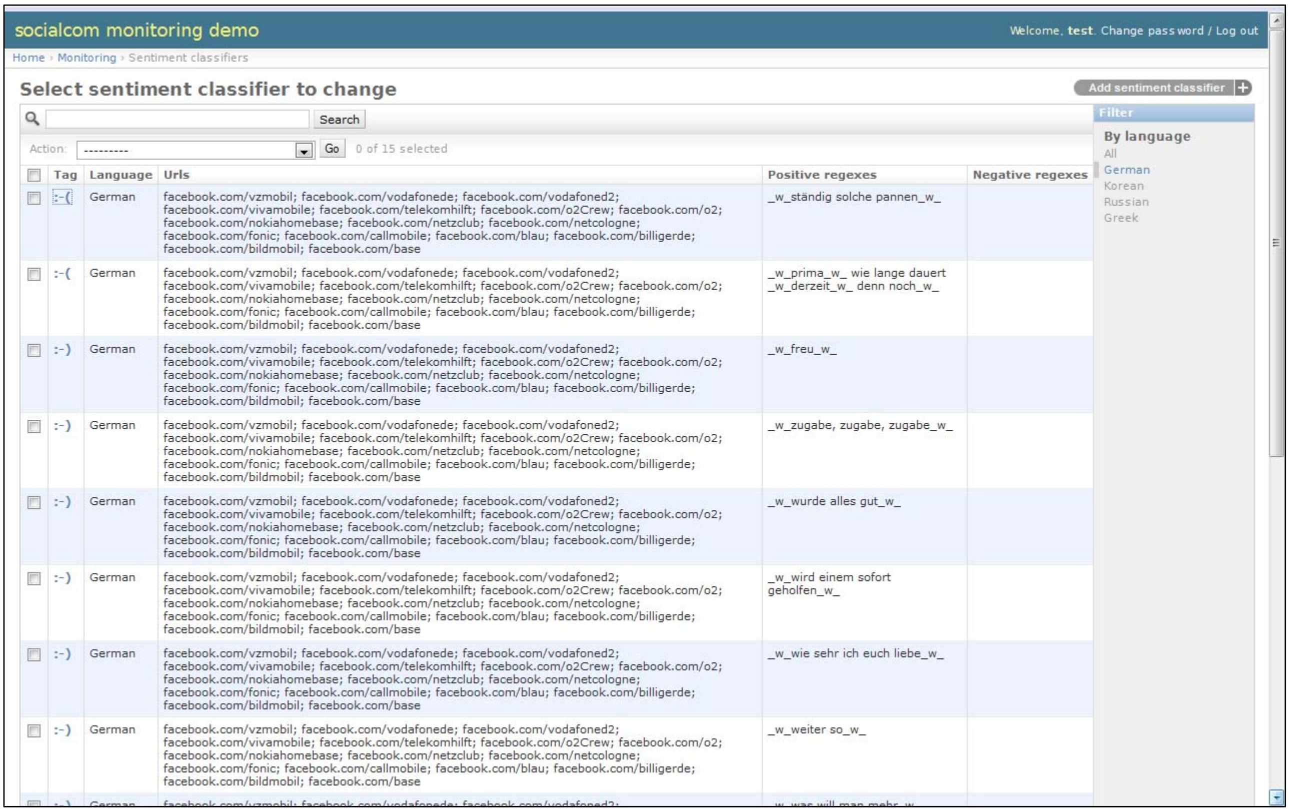


Figure 12. Sentiment classifiers in an SCM instance

A classifier has exactly one language and a set of URLs. It will only be applied to messages that have the same language and that stem from one of the URLs in the classifier's set of URLs. In general, content tags and product list are independent of each other. But many classifiers will have constraints that require that a product (or other entity) of a certain type be mentioned. Thus, a classifier that assigns the tag *phone available?* (e.g. to the message *When will the new iPhone be released?*) will probably include the mobile phone grammar in its constraints by using the special term `__mobile_phone__`.

SCM supports grammar creation in various ways: All objects in SCM, such as content classifiers, sentiment classifiers, countries, languages and messages are searchable and can be filtered by various criteria. Grammar experts usually deal with messages: They apply the content classifiers they (or other grammar experts) have created (or, for short, they apply *the grammar*) to all or certain messages. They can then filter the result: They might want to see only the messages that have received at least one content tag. Or they might want to see the messages that have not received a sentiment tag. Or they might want to see only Korean messages that have no content tag and are from a certain URL, and so on. Grammars and constraints that customer contact executives unknowingly generate can be supervised and manually corrected by grammar experts.

VII. DISCUSSION

A. No statistical approach

We think that the fact that customer contact executives can invent new tags and assign new or old tags to (badly) classified messages, if they mark the strings that are supposed to justify the assignment of the tag, is a good reason for not using a statistical approach. If we used a statistical approach, human work would be necessary at some point of the development process: Some algorithm would have to be trained.

In our approach, the human work is done in the customer management process. This way, two things are achieved in one step: The customer's request is answered and the classification algorithm is enhanced. An SCM instance is being enhanced while it is used. There is no need to interrupt the customer interaction in order to train SCM on new data that data specialists have created. Besides, manual intervention is much more straightforward and transparent, if a grammar of the type described above is used than it would be with a statistical algorithm. Our system is flexible in the sense that it can easily be modified in such a way that very specific requirements are met.

When a future user of SCM (a company that wants to interact with its customers) wants to assign every message that contains the word *hotline*, a certain tag such as *hotline problem*, then this requirement can be met by simply adding the line `hotline` to the positive constraints of the classifier called *hotline problem*.

B. Applying the DRY principle

SCM follows the DRY principle (**D**on't **r**epeat **y**ourself, see e.g. [10, p. 35]): Changes are only made in one place. An example: the Korean variants of the mobile phone name with the international ID *google-nexus-s* include: *google nexus s*, *google nexus-s*, *nexus s*, *nexus-s*, *구글 넥서스 에스*, *구글 넥서스에스*, *구글 넥서스 s*, *구글 넥서스-s*, *넥서스 에스*, *넥서스에스*, *넥서스* s, *넥서스-s*, *구글 nexus s*, *구글 nexus-s*, *google 넥서스에스*, *구글의 넥서스에스*.

This phenomenon is represented in SCM as follows: The Korean producer name corresponding to the international ID *google* has the variants *google* and *구글*. The Korean mobile phone name with the international ID *nexus-s* has the variants *nexus-s*, *넥서스에스* and *넥서스-s*. This is the only information SCM users have to store in order to make SCM generate these and many other variants. SCM generates *google nexus s*, *구글 넥서스 s* and similar variants using the general rule that in any permutation of a product name any minus character may be replaced by a space character. SCM generates *넥서스에스*, *넥서스 s* and similar variants using the general rule that the producer name may be omitted. And SCM generates *구글의 넥서스에스* using the two Korean variants of the producer name and the general rule that phone names can have the form `<producer name>의 <product name>`. (의 is a genitive affix, i.e. *구글의 넥서스에스* literally means *Google's Nexus S* or *Nexus S by Google*.)

We might, of course, add the general rule to SCM that any part of a Korean product name may be spelt either with Latin or with Hangul characters – according to several sets of transliteration conventions that are used in parallel.

Any change in a producer, tariff or product name object, such as the Korean mobile phone name with the international ID *nexus s* (cf. Fig. 13), has implications for the grammars of the message classifiers: Newly generated variants of the product name must be matched by all instances of `__mobile_phone__` in all grammars. For efficiency reasons, we compile all product names, tariff names, producer names, message classification grammars, sentiment classification grammars, and so on, into one single function.

This function is very efficient, because it doesn't do much more than apply one very large, compiled regular expression. The compiling and reloading of this function is done in the background, so the users of SCM do not need to know anything about it. They don't even have to understand the word compile. They just need to know that SCM sometimes needs a few seconds to be able to use changed objects.

We tried to design SCM such that it is easy for grammar experts to create and maintain grammars and similar entities (such as product names and producer names).

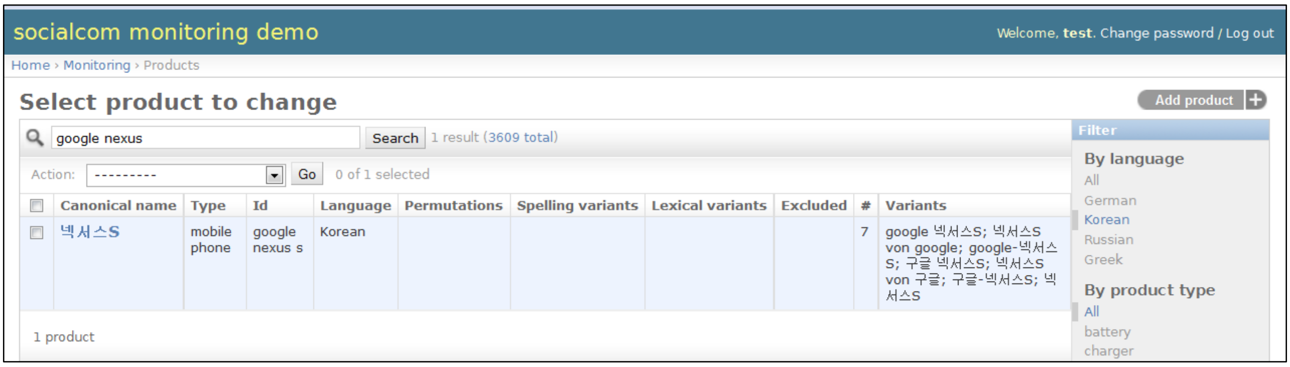


Figure 13. Korean sample of how to change the entry for the mobile phone *"google nexus s"*

We prefer long-term maintainability and usability over an algorithm that might perform better in a first demo version, but that is too rigid or complicated to be adjusted to new phenomena or to be otherwise maintained.

VIII. Conclusion and Directions for Future Research

In this article, we described a new technical service dealing with the integration of social networking channels into customer interaction management tools. We tried to show how these tasks can be automated to a large degree without making the system too complicated and intransparent. Our system is highly flexible and modular because it has a software agent for each of the entities involved in the monitoring and customer interaction process. In SCM, companies are mapped to company objects, products are mapped to product objects, text types are mapped to text type objects, and so on. Each of the objects can easily be modified by grammar experts. SCM makes sure that any change in an object is communicated to all the other objects that must know about it.

This research aimed to provide an initial understanding how effective communication between the business and its customers in an online social network can be handled. Mining social networks for classification purposes is no novelty; providing an assignment of customer messages to business processes instead of classifying them in topics did not exist before.

In the future, we hope to be able to enhance our system in several ways. Most importantly, SCM should be able to learn more things automatically. As many of SCM's grammars use special terms for matching certain types of expressions, such as `__mobile_phone__`, `__charger__`, `__headset__`, `__tariff__`, `__software__` or `__battery__`, minor code changes will suffice to enable SCM to find unknown mobile phone, charger or headset names in customer messages: As our grammars are designed for classifying messages of certain types, they contain very good contexts of named entities. Thus, a grammar that describes messages written by customers who want to install software on a certain mobile phone will probably contain good contexts for software and mobile phone names.

Future research will expand upon this study, investigating other social networking sites (apart from Facebook) and additional organisations across a range of non-telecommunication products or services. Even if our application produces the expected results, we are going to compare different feature extraction methods and classification algorithms in order to determine quality and efficiency in some experiments. Moreover, we intend to create a manually tagged test corpus of public customer posts on social networking sites that will be free for research purposes.

ACKNOWLEDGMENT

This work was supported by a grant from the German Federal Ministry of Economics and Technology. Our prototype was created in cooperation with the Munich-based German company Telenet GmbH Kommunikationssysteme.

REFERENCES

[1] SocialBakers.com, “Facebook statistics.” [Online]. Available: http://www.socialbakers.com/facebook-statistics/

[2] D. Yoon, S. M. Choi, and D. Sohn, “Building customer relationships in an electronic age: The role of interactivity of e-commerce web sites,” *Psychology and Marketing*, vol. 25, pp. 602–618, 2008.

[3] W. Gryc and K. Moilanen, “Leveraging Textual Sentiment Analysis with Social Network Modelling: Sentiment Analysis of Political Blogs in the 2008 U.S. Presidential Election,” in *Proceedings of the From Text to Political Positions Workshop (T2PP 2010)*, Vrije Universiteit, Amsterdam, April 9–10 2010. [Online]. Available: http://users.ox.ac.uk/_wolf2244/gryc moilanen t2pp 2010 final.pdf

[4] J. Krauss, S. Nann, D. Simon, K. Fischbach, and P. A. Gloor, “Predicting Movie Success and Academy Awards Through Sentiment and Social Network Analysis,” *ECIS European Conference on Information Systems*, 2008.

[5] J. K. Pal and A. Saha, “Identifying Themes in Social Media and Detecting Sentiments,” HP Laboratories, Tech. Rep. HPL-2010-50, 2010. [Online]. Available: http://www.hpl.hp.com/techreports/2010/HPL-2010-50.pdf

[6] G. Mishne, *Applied Text Analytics for Blogs*. Enschede, Netherlands: University of Amsterdam, April 2007.

[7] V. Jijkoun, M. A. Khalid, M. Marx, and M. de Rijke, “Named entity normalization in user generated content,” in *Proceedings of the second workshop on Analytics for noisy unstructured text data (AND ’08)*. New York, NY, USA: ACM, 2008, pp. 23–30.

[8] H. Raghavan and J. Allan, “Proper Names and their Spelling Variations in Automatic Speech recognition output,” University of Massachusetts, IR 361, 2005.

[9] R. Browne, E. Clements, R. Harris, and S. Baxter, “Business and consumer communication via online social networks: a preliminary investigation,” in *Australian and New Zealand Marketing Academy (ANZMAC) Conference*, 2009.

[10] M. Gross, “The Use of Finite Automata in the Lexical Representaion of Natural Language,” in *Proceedings of the LITP Spring School on Theoretical Computer Science: Electronic Dictionaries and Automata in Computational Linguistics*, M. Gross and D. Perrin, Eds. London: Springer-Verlag, 1989, pp. 34–50.

[11] ——, “Local grammars and their representation by finite automata,” in *Data, Description, Discourse, Papers on the English Language in honour of John McH Sinclair*, M. Hoey, Ed. London: Harper-Collins, 1993, pp. 26–38.

[12] ——, “The Construction of Local Grammars,” in *Finite-State Language Processing*, E. Roche and Y. Schabes, Eds. Cambridge, Massachusetts, USA: MIT Press, 1997, pp. 329–354.

[13] Z. S. Harris, “Mathematical Structures of Language,” *Interscience Tracts in Pure and Applied Mathematics*, vol. 21, pp. 152–156, 1968.

[14] ——, “Language and Information,” *Bampton Lectures in America*, vol. 28, pp. 33–56, 1988.

[15] R. Grishman, “Adaptive information extraction and sublanguage analysis,” in *Proceedings of Workshop on Adaptive Text Extraction and Mining at Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, USA, 2001.

[16] F. Guenthner, “Electronic lexica and corpora research at CIS,” *International Journal of Corpus Linguistics*, vol. 1, no. 2, pp. 287–301, 1996.

[17] B. Courtois, “Dictionnaires lectroniques DELAF anglais et franais,” in *Lexique, syntaxe et lexique-grammaire; syntax, lexis & lexicon-grammar*, C. Leclre, E. Laporte, M. Piot, and M. Silberztein, Eds. Amsterdam/Philadelphia: John Benjamins, 2004, pp. 113–123.

[18] D. Maurel and F. Guenthner, *Automata and Dictionaries*, Texts in Computing Science. London: King’s College Publications, 2006, vol. 6.

[19] S. Paumier, *Unitex User Manual 2.1*, 2010, http://igm.univ-mlv.fr/_unitex/UnitexManual2.1.pdf.

[20] S. Hunston and J. Sinclair, “A local grammar of evaluation,” in *Evaluation in Text: authorial stance and the construction of discourse*, S. Hunston and G. Thompson, Eds. Oxford, England: Oxford University Press, 2000, pp. 74–101.

[21] W. A. Woods, “Transition network grammars for natural language analysis,” *Communications of the ACM*, vol. 13, no. 10, pp. 591–606, 1970.

[22] S. Bsiri, M. Geierhos, and C. Ringlstetter, “Structuring job search via local grammars,” *Advances in Natural Language Processing and Applications. Research in Computing Science (RCS)*, vol. 33, pp. 201–212, 2008.

[23] D. McDonald, “Internal and external evidence in the identification and semantic categorization of proper names,” in *Corpus processing for lexical acquisition*, B. Boguraev and J. Pustejovsky, Eds. Cambridge, MA, USA: MIT Press, 1996, pp. 21–39.

[24] A. Mikheev, M. Moens, and C. Grover, “Named entity recognition without gazetteers,” in *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, 1999, pp. 1–8.

[25] N. Friburger and D. Maurel, “Finite-state transducer cascades to extract named entities in texts,” *Theor. Comput. Sci.*, vol. 313, pp. 93–104, February 2004.

[26] P. Murrell, *Introduction to Data Technologies*, Auckland, New Zealand, 2009.

[27] G. Friedrich and K. Shchekotykhin, “NameIt: Extraction of product names,” in *Sixth IEEE International Conference on Data Mining – Workshops (ICDMW’06)*, pp. 29–33, 2006.

[28] R. Colbaugh and K. Glass, “Estimating sentiment orientation in social media for intelligence monitoring and analysis,” in *ISI*, Vancouver, BC, Canada, 2010, pp. 135–137.

[29] B. Locke and J. Martin, *Named Entity Recognition: Adapting to Microblogging*. University of Colorado, 2009.

Michaela Geierhos received her PhD degree in computational linguistics, her MA degree in computer science, speech processing and computational linguistics from the University of Munich, Germany. She is currently graduate research associate and lecturer at Ludwig Maximilian University of Munich, Germany. Her research interests include linguistic information extraction, especially relation extraction, electronic lexicography and mining of short texts (in social networks, blogs, etc.).