KNOWNET 324408

Social Network Measurement Methodology and Tooling

Introduction

One of the objectives of the KNOWNET project is to measure effective knowledge transfer across the internal and external supply chain members in the insurance sector. As such it was imperative not only to develop a tool that was able to measure the impact of the different web based tools on the depth and breadth of knowledge sharing and learning, but also a device that allowed us to gauge participant engagement and motivation to sharing new ideas, insights and knowledge in a social supplier network.

The project has sought to devise a two pronged approach to measure and capture of knowledge transfer data and engagement in knowledge in sharing knowledge. In the first instance, quantitative data will be captured via logs and data mining tools. This stage of the measurement process will start as soon as the platform is launched. The second aspect of the approach will involve eliciting in depth qualitative data from participants via a series of interviews and questionnaires, either conducted on line, via a telephone conversation or in person. This data will focus on capturing more implicit knowledge flows, participant attitudes towards social media as a tool for sharing knowledge, ideas, insights and experiences, and finally, capture data on participant learning and implementation of new knowledge arising from engagement either directly or indirectly with the SSN platform.

The first part of this document describes the types of data measuring social activity and knowledge sharing, the KNOWNET consortium are looking to acquire in the project.

Data available for measuring social activity

Within the framework the data captured and that can be made available for further processing and analysis can be split into two main classes:

- Standard web server logs
- Social graph properties ford processing and offline analysis

The standard web server logs capture the different web requests made to the KNOWNET framework. They are useful for deriving basic statistics like number of daily requests, request distribution through the day/week/month, etc… We use the standard apache log format for storing the web server logs.

Social network analysis methods can be used to study the properties of the social graph. Related content analysis methods can be used to understand the relations between different items of content. Most if not all of these methods are computationally heavy and suitable for offline processing.

We can export data for offline processing in both csv and json file formats. Currently this has to be a manual process, as there are too many variations possible. If we discover often used projections of data, we shall automate the process.

Measurement and analysis process

The process, derived from the classic Data Mining (DM) and Knowledge Discovery in Databases (KDD) processes, consists of the following steps (see also Figure 1):

1. Selection
2. Pre-processing
3. Anonymisation
4. Transformation
5. Analysis
6. Interpretation/Evaluation

The first three steps are described in more detail in this document, while the transformation, analysis and evaluation steps are discussed in detailed in the KNOWNET - Social Networking Analysis Manual

The SELECTION STEP is used to extract the data we are interested in from the available sources. In the KNOWNET case, the available sources are the KNOWNET database and log files, as described in more detail later in this document.

Example selection queries:

- select all content tagged with the ‘brilliant’ keyword
- select all users who have posted content tagged with #claims
- select all users who follow users posting content tagged with #claims

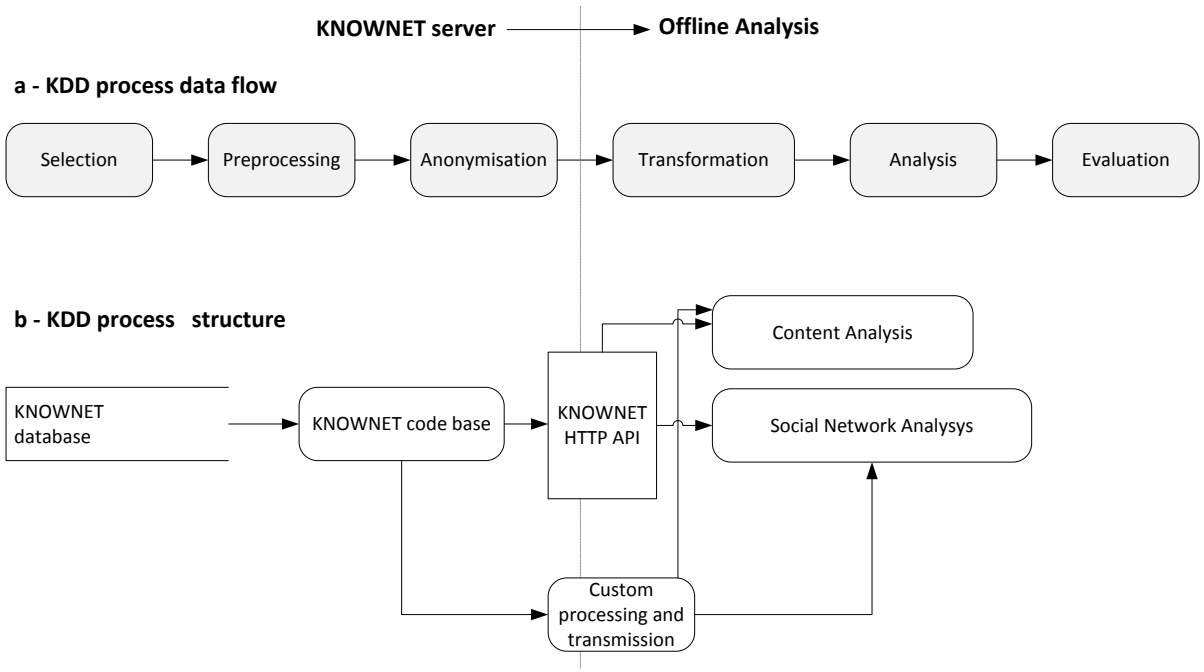

Figure 1 KNOWNET measurement and analysis process modelled as a Knowledge Discovery in Databases (KDD) process
a – KDD process dataflow diagram; b – KDD process structure
The diagram illustrates the how process phases and system structure map to location – where the operations are performed, i.e server or offline on an expert workstation or equivalent

PRE-PROCESSING is used to filter and/or veto malformed, unrelated or otherwise unusable data. This step is often partially fused with the previous, selection, step. Usually it is very specific, heavily dependent to the data set quality and content.

Example pre-processing filters:

- remove all content generated by the system's admin user
- remove all content tagged with #joke
- remove all root or conversation start content without hashtags and its replies

ANONYMISATION is a required step in our system – all user recognisible or business critical content markers should be removed before the data is moved off-line for processing. For example: "substitute user names for some user ids, preserving the user-content relations"

TRANSFORMATION is a process to convert the data between different formats, it could include data aggregation, etc…

ANALYSIS – data analysis or data mining – using statistical or other methods try to gain insights of the collected data

INTERPRETATION AND/OR EVALUATION of the analysis results is the most important step of the process, it allows us to summarise the insights gained so far in the process.

This is an adaptive, often repetitive, data and knowledge exploration process, as we rarely know the answers upfront, and usually new answers help formulate new questions.

The transformation, analysis and Interpretation stages of the process are described in the KNOWNET - Social Networking Analysis Manual

Selection and pre-processing

Framework metadata

In the framework database we capture the following interesting entities:

- **User** – an active person in our social network
- **Group** – a collection of users, can be used to associate and optionally restrict content to a particular group of people
- **Document** – Micro-post, blog post, wiki page, …
- **Tag** – Keyword, hash-tag

Among others, we capture the following interesting relations:

- **Direct user-user relation** – designed to capture strong social ties like: follows, friend of, …
- **Group membership**, user-group, relation
- **User – Document relation** – designed to capture properties like: author of, likes, dislikes, mentioned in, etc…
- **Document – Tag** relation – answers the question what are the keywords associated with a document
- A number of other, less important, but useful relations between tags, documents, groups and users

The above relations can be used to approximate and reconstruct the social and content graphs by constructing derived, more complex relations by relation composition. This is usually performed on database level, for example using SQL joins.

As a simple example let's get an approximation of the social graph based on the 'belongs to the same group' tie. For simplicity, we will use a binary connected, not connected model, rather than a weighted graph. We have the set of groups G, users U, and the binary relation BelongsTo (u,g) – if a tuple exists in the BelongsTo relation then the user u belongs to the group g. That is usually expressed in SQL directly, for example:

```
SELECT * FROM UserGroup AS ug1
INNER JOIN UserGroup AS ug2 ON ug1.group = ug2.group
WHERE  ug1.user != ug2.user
```

Another example, would be the ‘related by content’ class of derivative relations. Let’s look at the “created content with the same keywords”. This is a composite relation, composing the author of content (UserDoc relations, tagged with the author class) relation, with DocumentTag relation, where the tags are equal. The SQL should be something like:

```
SELECT * FROM UserDoc AS ud1
INNER JOIN DocTag  AS dt1 ON ud1.doc=dt1.doc
INNER JOIN DocTag  AS dt2 ON dt1.tag=dt2.tag
INNER JOIN UserDoc  AS ud21 ON ud2.doc=dt2.doc
WHERE ud1.user != ud2.user
```

We can study such social graph approximations in isolation or blend different ones together into composite models, for example we could calculate the union of the above two result sets to create a new model.

Direct user – user relations

The simplest relation which can be extracted from the database is the user-user relation. As it is a many-to-many relation, it lives in a separate database table (UserUser). It captures Follows, FollowedBy, Friend of and similar types of social ties. The diagrams on Figure 2 illustrate the graph simplification process for this type of social meta-data.

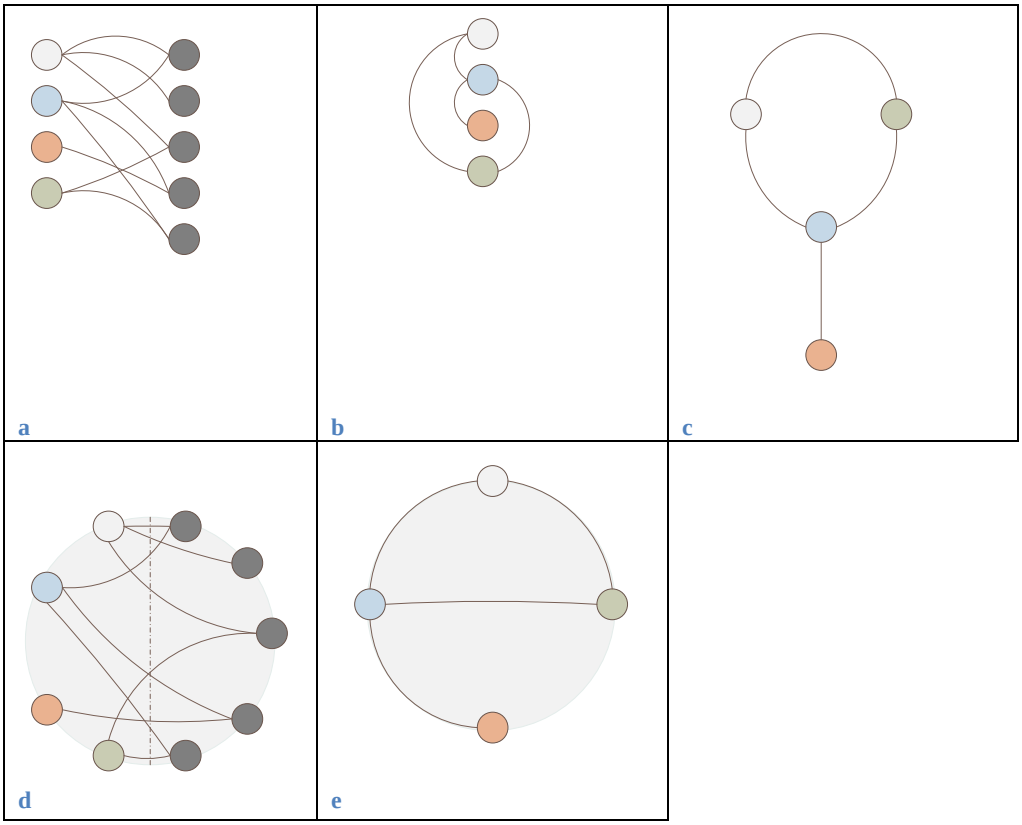

Figure 2 User – User relations diagrams and derivations
a – user – user relations bi-graph; b & c – simplified user relations graph; d – user – user relations chord diagram; e – simplified user – user relations chord diagram
Note: in all diagrams the dark grey nodes represent a relation captured in the database, the colour nodes represent individual users

The diagram on Figure 2 (a) illustrates the structure of the relation as it lives in the database. The dark-grey nodes of the graph represent the tie data in DB (the UserUser table). The graphs on Figure 2 (b,c) represent the simplified graph, where the relation data is substituted by graph arcs, the only difference between the diagram is layout.

Sometimes, when studying complex graphs, it may be useful to use chord diagrams – the start and simplified user-user example can be seen on Figure 2 (d,e)

Indirect user – item – user relations

A slightly more complex class of relations are the user-item-user relations. Relations where some items – either content, for example co-authors or groups act as intermediaries labelling a social tie.

Let's look again at the 'users belonging to same group' social tie. The process of deriving the relationship graph is illustrated on Figure 3. It is similar to the process in the previous section, but has an extra step at the start.

It the beginning we have a tri-partite graph (trigraph) with nodes corresponding to users (left), groups (right) and the database relation nodes. If we close over the groups, we get the simplified version of Figure 3 (b), which is of similar complexity to the graph of the user-user relation. The next steps eliminate the intermediate, non-user nodes of the graph.

Similarly, using the same set of diagrams, we could illustrate the derivation of user ties based on co-authors or friend of friend relations.

We could derive more structurally more complex graphs. Some examples of interesting social ties based on:

- **User-Document-Tag-User** Authored content with the same keyword(s)
- **User – User –User** Friend of a friend like relations (2^{nd} degree acquaintance)
- **User – Tag – User** Users self-tagged with the same keywords
- Weighted graphs of any of the kinds above

The particular relations we could be interested in depend on many factors, including the social network itself, the context and the particular properties of the social network we would like to study. As such, this requires an iterative, exploratory approach.

The examples given above illustrate activities during the selection and pre-processing stages of the KDD pipeline.

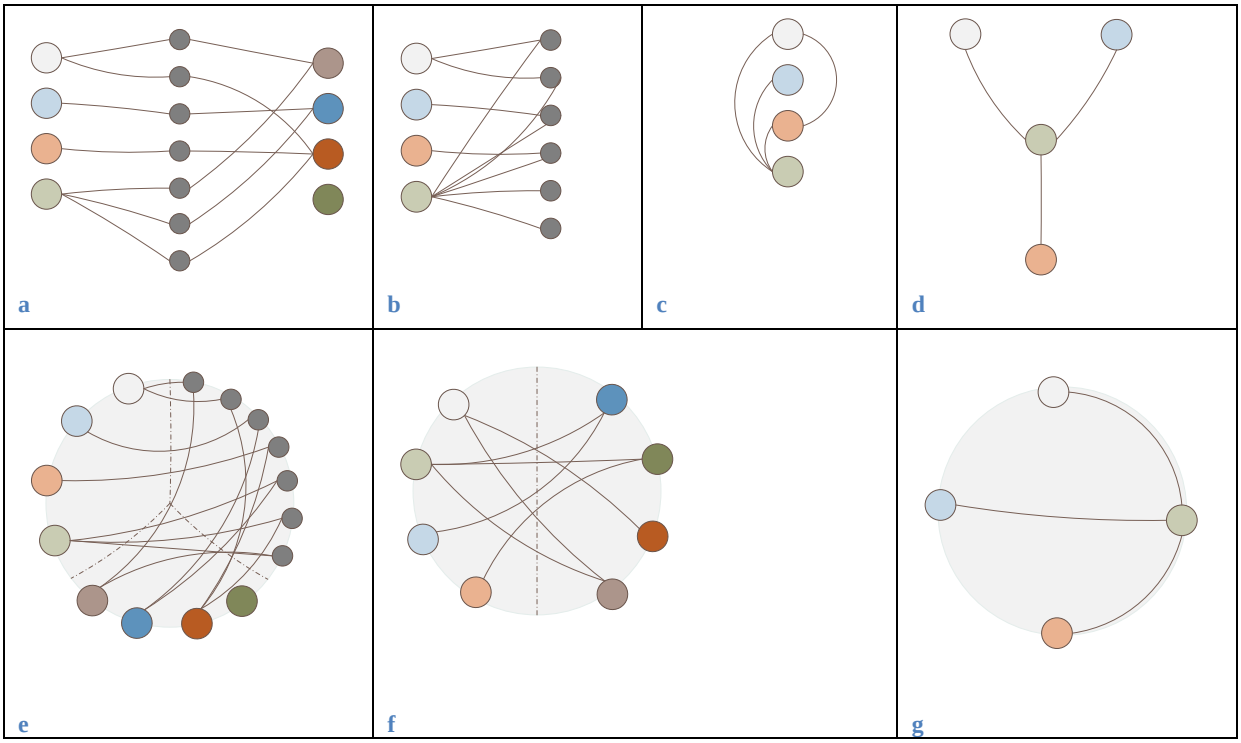

Figure 3 User – Item - User relations diagrams and derivations
a- user-item relations trigraph; b- derived user – user relations bigraph; c & d – simplified user-user relations graph; e – user-item relations chord diagram; f – simplified user-item relations chord diagram; g – derived user-user chord diagram
Note: in all diagrams: the dark grey nodes represent a relation captured in the database; the dimmer coloured nodes represent individual users; the brighter coloured nodes represent individual items

Anonymisation

We have two competing constraints – we must not reveal any user data like usernames or emails, we have to preserve as much structure as possible for user at the later stages of the KDD process.

Naive randomisation of user and document names and ids won't work, as we lose the social and content graph data. We have to opt for a weaker form – we shall use only user, document, tag, and other entity ids. The original full names and content are recoverable, with access to the original data. The threat of revealing sensitive data is minimised, as we intend on exporting only metadata - user, or document, or author, or tag ids - without exporting for analysis content, names and other potential sensitive information

When exporting content, we would have to implement strict filtering procedures, in order to minimise the risks of sensitive information exposure. This approach allows for maintaining reasonable privacy while exposing the rich social and content network structure

Content analysis

Social network analysis, i.e. reconstructing the social graph is not enough. In order to understand the knowledge and the associated structures captured by the KNOWNET framework, we have to have an understanding of the content graph, how different pieces of content relate to each other.

Similar to the social network analysis (SNA) case, we have direct, item to item, and indirect relations. The direct relations are captured in the database in the DocDoc table as the triple (Document, Document, Type), where Type is the type of the relation, for example reply, comment …

For capturing similarity and distance between documents, we can construct a number of indirect, transitive relations, a process related and similar to SNA (see Figure 2 and Figure 3), using straightforward re-mapping of node colours from Users to Items, and fresh mappings of the secondary nodes of the corresponding graphs on the diagrams:

- **Item-User-Item** – for example, authored by the same person, using composed or chained UserDoc relations
- **Item – Tag – Item** – items tagged with the same keyword, using composed DocTag relations
- **Item – Group – Item** – items in the same group, using the group membership relation GroupDoc

Of course we could calculate other, composed or 'deeper', more complex indirect relation graphs, include time difference in the partial order calculations, etc… The information flow analysis, similar to SNA, is based and builds on graph analysis and transformation, and shares a number of common metrics with SNA, like range, betweenness, density, distance, etc…

We can explore a menu of collaborative filtering and content recommendation algorithms, the choices will have to be made when we have sufficient data for analysis. A short list of some open source tools available:

- APACHE MAHOUT – scalable machine learning library (http://mahout.apache.org)
- NLTK – python based natural language processing toolkit (http://nltk.org)
- APACHE PIG - high-level data-flow language and execution framework for parallel computation. (http://pig.apache.org)
- APACHE SPARK - A fast and general compute engine for Hadoop data, which supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation. (http://spark.incubator.apache.org/)
- GEPHI - an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. (https://gephi.org/)
- R - a free software environment for statistical computing and graphics. (http://www.r-project.org/)

Qualitative measures

In order to gain greater depth on the types of knowledge (implicit, and explicit) that is being or has been exchanged between individuals and groups as well as unpack the relationships we could not predict from the data, we will be complementing the quantitative measures and finding with interviews. Social media treats all users the same, trusted friend or total strange with little or nothing in between. In reality, relationships fall everywhere along this spectrum. The interviews will complement data from quantitative measures to confirm and strengthen results. Additionally, we will be able to more fully understand when and where knowledge in all its forms has been exchanged and acted on. Qualitative data collection allows us to also examine the attitudes, feelings and follow up behaviours of participants that arise from engaging with the platform and which cannot be captured using the tools listed above.

Interviews will be conducted on line, over the phone or in person some months after the platform has been up and running. Interviews will be semi structured with opportunities for respondents to provide examples of implicit/tacit knowledge exchanges.

Tacit knowledge cannot be captured in the same way as explicit knowledge. Tacit knowledge could be classified into two dimensions: the technical and the cognitive dimension (Herrgard 2000). The technical dimension encompasses information and expertise in relation to 'know-how' and the cognitive dimension consists of mental models, beliefs and values (Gore & Gore 1999). The literature reveals two fundamentally different and competing schools of thought regarding diffusion and codification of such knowledge. One believes that tacit knowledge can and must be made explicit for sharing and the other regards tacit knowledge as always being tacit.
Polanyi (1958) sees tacit knowledge as a personal form of knowledge, which individuals can only obtain from direct experience in a given domain. According to Polanyi (1958), this knowledge is held in a non-verbal form, and therefore, the holder cannot provide a useful verbal explanation to another individual. Moreover, as he contends, tacit knowing is such an elusive and subjective awareness of the individual that it cannot be articulated in words. It is from Polanyi's argument that the differentiation between tacitness and implicitness was apparent, and from his terminology, tacitness was evidently different from implicitness. Implicitness, another form of expressing knowing, does exist. It implies that one can articulate it but is unwilling to do that because of specific reasons under certain settings such as, intrinsic behaviour in perception, cultural custom, or organisational style (Li & Gao 2003). Therefore, by describing implicit knowledge, Polanyi was referring to the technical dimension of the tacit knowledge, whereas cognitive dimension purely represented the tacit knowledge that he considered as always being tacit.
"...we classify human knowledge into two kinds. One is explicit knowledge, which can be articulated in formal language including grammatical statements, mathematical expressions, specifications, manuals, and so forth.... A more important kind of knowledge is tacit knowledge, which is hard to articulate with formal language. It is personal knowledge embedded in individual experience and involves intangible factors such as personal belief, perspective, and the value system" (p.viii).

For these reasons, and the difficulty in capturing implicit (and tacit) knowledge, an important part of the data collection process will involve in depth semi-structured

interviews. This will enable the KNOWNET team to gain an understanding from the participants as to whether and what type of knowledge has been exchanged – whether it be explicit or difficult to articulate in formal language, but which may be apparent through story- telling, and experiences of the participants in their exchanges on the platform.

Summary

The KNOWNET framework database is designed to capture rich meta-data about people, content and their relations. The meta-data is the basis for calculating approximations of the social and content graphs in order to perform social network and content analysis. The results are anonymised and delivered for analysis in json, csv, or apache log format. The latter is used only for traditional web server logs.

Additionally, the capture of content(story telling etc) and attitudes towards engagement with the SSN platform will be conducted using qualitative data collection techniques.

References

Polanyi, M. (1958) Personal Knowledge Towards a Post-critical Philosophy. Routledge and Kegan Paul Ltd, London

Herrgard, T.H. (2000) Difficulties in the diffusion of tacit knowledge in organisations. Journal of Intellectual Capital, Vol. 1, No 4, pp. 357-365.

Gore, C. and Gore, E. (1999) Knowledge management: the way forward. Total Quality management, Vol. 10, No 4-5, pp. 554-60.

Li, M., and Gao, F. (2003) Why Nonaka highlights tacit knowledge: a critical review, Journal of Knowledge Management, Vol. 7, No 4, pp. 6-14.