DOI: 10.2298/CSIS120920052V

Forecasting the Acceptance of New Information Services by using the Semantic-aware Prediction Model

Luka Vrdoljak[1], Vedran Podobnik[2] and Gordan Jezic[2]

[1] Intesa Sanpaolo Card Ltd
Radnička cesta 50, Zagreb, Croatia
looka.vrdoljak@gmail.com

[2] University of Zagreb
Faculty of Electrical Engineering and Computing
Department of Telecommunications
Unska 3, Zagreb, Croatia
{vedran.podobnik; gordan.jezic} @fer.hr

Abstract. With the constantly increasing competition on the information service market, service providers should enhance their business processes by introducing new mechanisms. These novel mechanisms must include almost real-time detection of business opportunities (as well as possible failures), necessary resource prediction, and finally profit forecasting. Presented challenges can be tackled by using growth models for service acceptance prediction. However, common growth models have certain shortcomings when it comes to forecasting consumer interest in new services. Two main shortcomings are: i) limited precision; and ii) a short, but yet existing, time delay. Possible solution that minimizes the specified shortcomings is semantic reasoning, which can be used for detecting similarities between services already on the market and ones that are just to be introduced. Consequently, it becomes possible both to increase forecasting precision and eliminate time delay caused by the need to collect a certain amount of data about the new service before any prediction can be made. Our approach, the semantic-aware prediction model, can thus replace the common subjective service similarity approximation approach. Elaboration and verification of the semantic-aware prediction model are conducted on a case of forecasting YouTube clip popularity.

Keywords: Consumer Relationship Management, Consumer Managed Relationship, Forecasting, Growth Models, Semantic Reasoning, YouTube.

1. Introduction

With the constantly increasing competition on the information and communication market, service providers must focus on maintaining

consumer satisfaction. In order to do so, service providers must observe their consumers individually, rather than seeing them just as a part of a certain market niche [1]. Such individual and personalized approach can be recognized on the market through two most common concepts: *Consumer Relationship Management* (CRM) and *Consumer Managed Relationship* (CMR) [2][3]. These two concepts reside on three basic ideas: i) consumer experience management (CEM), ii) real-time analysis, and iii) technology used for cost decreasing and creating a consumer-oriented environment [4][5][6].

Most companies react only when a number of consumers decreases (i.e., churn rate dominates over growth rate). However, by then it is usually too late to intervene. On the other hand, using *Predictive Analysis* (PA) can lead to a proactive consumer retention strategy [5]. By analyzing consumer habits, expenditure and other behavior patterns, forecasting models can determine the probability of a decrease in consumers' interest in a certain service, or even the potential interest in a service that has yet to be introduced in the market [7]. Additionally, predictions could be improved with the use of *Semantic Web* technologies which enable detection of similar services already on the market [1][8].

In this paper, we first make an insight in the common forecasting models and state-of-the-art technologies for semantic service profiling (Section 2). Then our proposed system is presented through key processes (Section 3) and its architecture (Section 4). In Section 5 we conduct an evaluation of our ideas on YouTube clip popularity forecast. Section 6 concludes the paper and presents the planned future research.

2. Related Work

After a long era of easily predictable fixed voice telephone services, information and communication industry has come to a period of intensive introduction of a very wide spectrum of numerous new services [9]. Rapid technological development and liberalization have made the information and communication market a very dynamic environment where forecasting is becoming increasingly important. By understanding data patterns during information and communication services' life-cycles, a service provider can perform optimal business planning of its capacities, investments, resources (e.g. human potentials and equipment), marketing and sales. However, there is always the problem of bridging the gap between collected historical data and the anticipated value in the future due to the lack of reliable input data for forecasting.

2.1. Forecasting in Information and Communication Industry

Forecasting is a permanent process in which all new information and changes on market have to be taken into account for business planning and enhancing business performance [9]. Nowadays, timely implementation of newly acquired knowledge in business processes represents one of the extremely rare competitive advantages.

During its life-cycle, every service passes through certain phases. Understanding and forecasting of these segments in a *service life-cycle* (SLC) for the business planning purposes are becoming increasingly important in competitive market environment, especially for services resulting from emerging technologies, such as information and communication services [9]. In general, scope of information and communication service forecasting could be defined as set of techno-economic indicators forecasting necessary for developing business case in the industry. For information service providers it usually consists of:

i) consumer growth forecasting,
ii) market share forecasting,
iii) volume - pricing forecasting,
iv) *average revenue per user* (ARPU) forecasting, and
v) forecasting of revenue in total.

According to the available literature, software tools and the general experience, when it comes to forecasting in information and communication industry the following methods are used most often [9]:

i) new service acceptance forecasting by using growth models (e.g. the logistic and the Bass growth model),
ii) forecasting models based on seasonal variations elimination and autoregression (e.g. exponential smoothing and the Box-Jenkins method),
iii) cross-section models for the forecasting based on the relations between different services or the relations between equal services in different markets,
iv) scenario methods, and
v) Monte Carlo for revenue, costs and net present value (NPV) forecasting.

This paper will focus on consumer growth forecasting for new services using growth models.

2.2. Common Service Growth Models

The life-cycle (SLC) of every service consists of phases shown in Fig. 1 [10]: *development*, *introduction*, *growth*, *maturity* and *decline*. A typical service during its life-cycle passes through specific phases of market adoption, which

can be observed through the number of service users. Understanding these phases in a SLC is especially important for highly competitive market environments and particularly for services based on emerging technologies. Numerous researches have resulted in a conclusion that these phases can be described by mathematical models which will be briefly explained here.

Growth models mathematically describe patterns of growth in nature and economy, illustrate how a certain environment reflects on the growth, as well as enable future growth forecasting. Particularly, diffusion of new ideas and technology, market adoption of new products and services, as well as allocations of restricted resources has characteristic *S*-shaped (sigmoidal) growth. Most commonly used *S*-shaped models for initial phases of a SLC (i.e. introduction, growth) are: *logistic model*, *Bass model*, and *Richards model*. Later phases of a SLC require more complex models (e.g. *Bi-logistic growth model*) [9]. This paper will be focused on the services in their initial SLC phases.

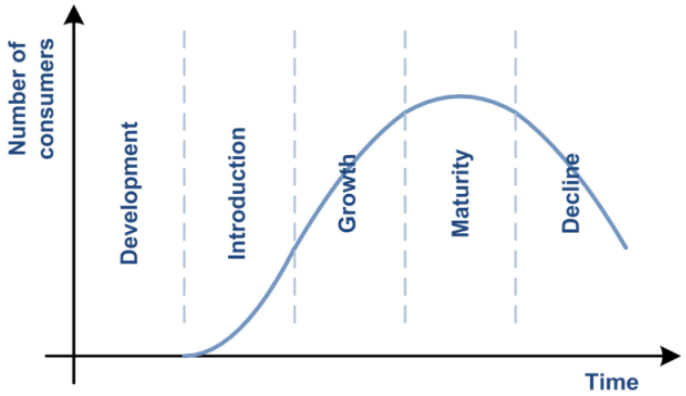

Fig. 1. Information and communication service life-cycle.

Logistic model. The logistic model $\mathrm{L(t)}$ is best used for describing growth of the number of service consumers in time in a closed market, isolated from other services. The model is defined with three parameters: M– market capacity, a – growth rate, and b – time shift parameter, as is shown in (1) [11].

$$L(t; M, a, b) = \frac{M}{1+e^{-a(t-b)}} \tag{1}$$

The logistic model is a widely used growth model with numerous useful properties for technological and market development forecasting. During the first phase, growth of the logistic model is exponential, but later negative feedback slows the gradient of growth as the number of consumers approaches the market capacity limit M. Individual impact of each parameter can be seen in Fig. 2.

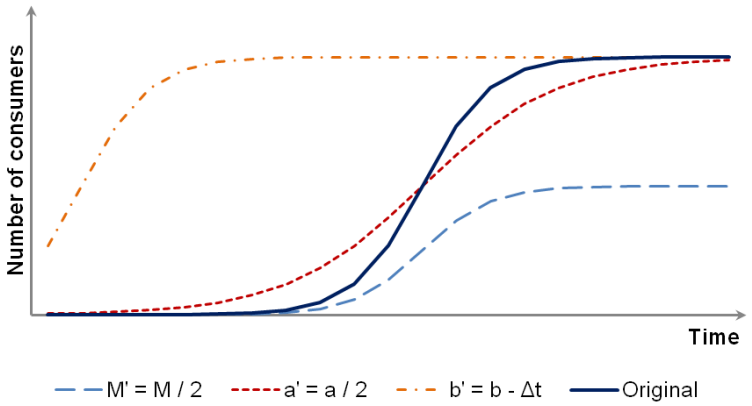

Fig. 2. Interest in a service described by the *logistic model* with different parameters.

Bass model. The most common model for describing new service diffusion is the Bass model [12]. The Bass model $\mathrm{B(t)}$ corrected the deficiency of simple logistic growth (i.e., slow growth which cannot be applied to services that have instant growth after market introduction, and no point where $\mathrm{L(t)}$ equals zero) by taking into account the effect of innovators via coefficient of innovation p. The model divides a population of M adopters in two categories [13][14]:

i) *innovators* (with a constant propensity to purchase), and
ii) *imitators* (whose propensity to purchase is influenced by the amount of previous purchasing).

Bass diffusion model is defined by the following four parameters:

- M – market capacity;
- p – coefficient of innovation, $p > 0$;
- q – coefficient of imitation, $q \geq 0$, and
- t_s – moment of service introduction, $B(t_s) = 0$.

These parameters define the model as shown in (2). The Bass model has a shape of S-curve, as does the logistic model, but the curve is shifted down on the *y*-axis. Fig. 3 shows the effects of different values of parameters p and q on form of S-curve, with fixed values for M and t_s [9].

$$B(t; M, p, q, t_s) = M \frac{1 - e^{-(p+q)(t-t_s)}}{1 + \frac{q}{p} e^{-(p+q)(t-t_s)}} \quad (2)$$

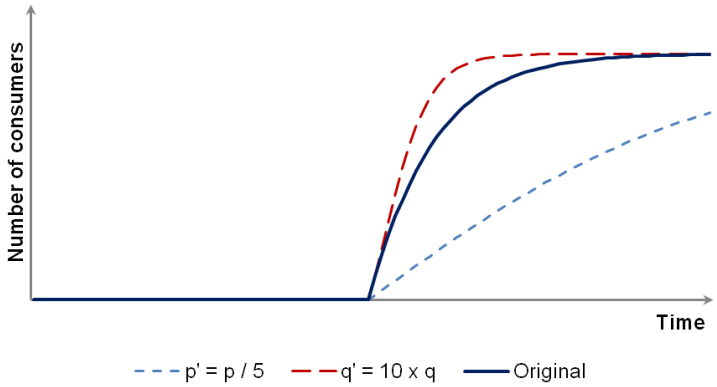

Fig. 3. Interest in a service described by the *Bass model* with different parameters.

The Bass model is widely used for the long-term forecasting of a new service market adoption when interaction with other services can be neglected. Furthermore, the Bass model can be used when limited or no data is available (i.e. market adoption forecasting prior to service launch), which is also a focus of our paper.

Forecasting based on growth models. In order to use growth models for service growth forecasting, it is necessary to implement growth model parameter determination mechanisms. When these mechanisms rely on mathematical methods, parameters for a model with k parameters can be determined if there are at least k known data points [9].

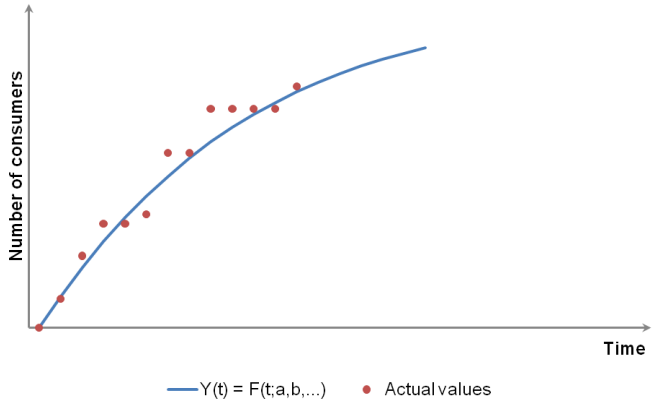

Fig. 4. Transforming actual data points into a model defined as $Y(t)$.

However, the problem occurs when the number of known data points is insufficient, or just too low, so the prediction for service growth results in a wide interval of possible values. While best practices in such cases recommend obtaining model parameters via subjective judgmental assumptions and placing them in optimistic-pessimistic intervals we propose a different, more objective approach. Namely, researches have shown that services with similar characteristics tend to have similar parameters when their growth is represented with the Bass (or logistic) model [15]. Taking this into account we propose using semantic reasoning to enable computers to autonomously calculate the level of similarity between any pair of services, which can then be used for more objective growth model parameter determination.

2.3. Semantic reasoning

Semantic reasoning has proven to be a very efficient method for resource description and matchmaking. Using various query languages, based on *Structured Query Language* (SQL) syntax, it is possible to perform semantic data extraction, thus enabling efficient matchmaking of service profiles once they have been created according to a certain standard. Such matchmaking enables service comparison to be performed according to true, semantic similarities, rather than keyword matchmaking (e.g. YouTube clip keywords).

Semantic service profile. Semantic reasoning necessary for information service description and ultimately service matchmaking, can be based on existence of two types of information: an ontology that provides a generic structure of concepts (e.g. *DBpedia Ontology*) used for resource annotation (e.g. service profile), and service profiles which are formed as sets of attributes and values, where each of these values can have a different type (e.g. number, instance within the ontology, plain text, Boolean) [8].

Semantic matchmaking. Semantic matchmaking algorithm uses semantic web languages in order to determine similarities between resources according to true, semantic meaning, rather than syntactic structure [16]. The algorithm must be capable of expressing the similarity between two resources as a *numeric value*, not just a binary value defining whether the resources are completely matching. Semantic matchmaking must, in a way, assign a numeric value to the following elements of a semantic resource description [8]:

- Position within the ontology class hierarchy (e.g. multimedia content, mobile application) – the distance between classes the two resources instantiate indicates the probability of their semantic similarity. The

smaller the distance between them, the numeric value defining their similarity should be higher. Designing the ontology so it reflects the relation between two concepts ensures that a significant part of the resource description is contained in the instantiating itself;
- Attributes that take on values with common data types – it is necessary to establish rules on how to compare binary values, integers, decimal numbers and textual values. Binary values can be rated with 0 or 1, depending on their match. The same rule can be applied to textual values. When it comes to integers and decimal numbers, ratio of the lower and the higher value can represent the similarity between two values;
- Attributes with class instances as values – in this case the approach should be the same as the one when comparing the position within the ontology.

The algorithm should also enable defining the relevance of each attribute, or set of attributes. Final resource similarity would then be calculated as a function of individual attribute similarities and their relevance factors (e.g. weighted arithmetic mean), and the position of the very resources within the ontology.

3. Service modeling system

In order to perform new service growth prediction it is necessary to ensure mechanisms for four basic processes:

i) creating semantic profiles based on service description,
ii) mapping existing services' historical data into growth models (e.g. Bass or logistic model),
iii) comparing newly introduced service with existing services, and finally, and
iv) calculating newly introduced service growth model and corresponding parameters.

3.1. Semantic profiling

In order to perform service matchmaking, it is necessary to define a generic semantic service profile structure. Such structure should include information about the type of information service, the content provided by the service, technical characteristics of the service, service acceptance data, and growth model data derived from the acceptance data. Such generic structure can be described as:

$$p_{s_i} = (i_{s_i}, c_{s_i}, t_{s_i}, a_{s_i}, m_{s_i}) \quad (4)$$

where service profile p_{s_i} consists of five parts:

- i_{s_i}- Identification information (e.g. type of service, unique identifier);
- c_{s_i}- Content information (e.g. category, keywords);
- t_{s_i}- Technical characteristics (e.g. multimedia clip length);
- a_{s_i}- Acceptance data (e.g. number of multimedia clip viewers or comments);
- m_{s_i}- Growth model data (e.g. chosen growth model, model parameters).

3.2. Modeling existing services

Number of consumers of a certain service, when observed through time, forms a rather irregular set of discrete data points. Our system must transform this stochastic set of data into a smooth *S*-curve that approximates the actual numbers with a satisfactory degree of deviation (Fig. 5). Finding the correlation is performed in two steps: recognizing the correct model (e.g. Bass or logistic model) and calculating the corresponding parameters. As was mentioned earlier, a data model defined by k parameters requires at least k known data points. Services with considerable time on the market will normally have more than k known data points and that is where the weighted least squares method comes to use [9].

Weighted least squares method. The method of least squares is a common approach for approximation of overdetermined system solution (i.e. sets of equations in which there are more equations than unknowns). The overall solution minimizes the sum of the squares of the errors made in the results of every single equation. The best approximation in the least-squares sense minimizes the sum of squared residuals, a residual being the difference between an observed value and the fitted value provided by a model.

Least squares problems are divided into two categories: *linear* (or ordinary least squares) and *non-linear least squares*, depending on whether or not the residuals are linear in all unknowns. The linear least-squares problem can be solved in a finite number of standard operations. The non-linear problem is usually solved by iterative refinement; during each iteration the system is approximated by a linear one, thus the core calculation is similar in both cases.

In some cases the observations may be weighted as they may not be equally reliable. Weighted least squares method is very efficient when it comes to small data sets. The main advantage of weighted least squares method over other methods is the ability to handle regression situations in which the data points are of varying quality. If the standard deviation of the random errors in the data is not constant across all levels of the explanatory variables, using weighted least squares with weights that are inversely

proportional to the variance at each level of the explanatory variables yields the most precise parameter estimates possible [16].

The biggest disadvantage of weighted least squares is probably the fact that the theory behind this method is based on the assumption that the weights are known exactly. This is almost never the case in real applications, of course, so estimated weights must be used instead. It is important to remain aware of this potential problem, and to only use weighted least squares when the weights can be estimated precisely relative to one another [17].

3.3. Introducing and modeling newly introduced services

The final goal of the proposed system is to predict consumer interest in newly introduced services by calculating its growth model parameters. In order to achieve that it is firstly necessary to see where the newly introduced service fits in the existing set of services on the market. We propose using semantic matchmaking algorithm described earlier to detect similar services [8]. Once similar services have been identified, it is possible to detect the most appropriate growth model (e.g. the most common growth model among similar services) and calculate the chosen model parameters based on parameter values of the similar services, taking corresponding semantic similarities as weight factors (i.e. the most similar services are the most important while calculating parameter values) [1]. This is shown in Fig. 5, where p_{s_i} represents the semantic profile of service i, ss_{ni} the semantic similarity between the new service n and existing service i, and m_{s_n} the model data of the new service.

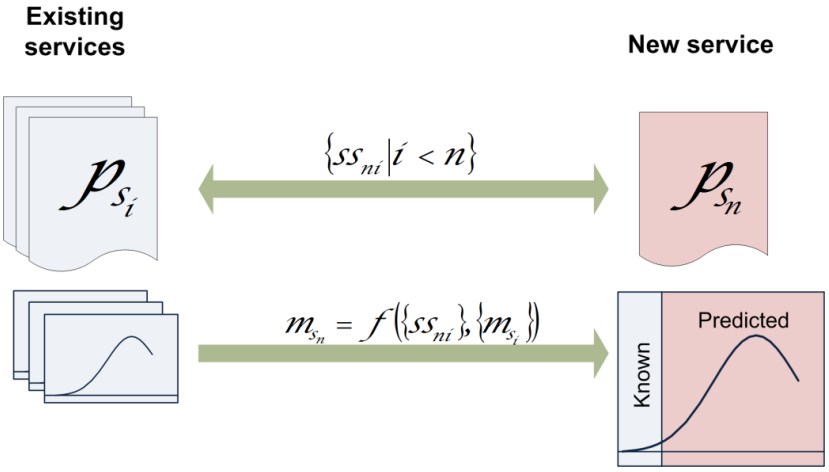

Fig. 5. The process of forecasting consumer interest in a newly introduced service.

4. Forecasting system architecture

The generic architecture of the forecasting system we propose in this paper consists of three main entities (Fig. 6[1]):

i) one or more service providers,
ii) semantic repository, and
iii) forecasting application.

4.1. Forecasting system entities

The *service provider* (e.g. multimedia content providers, mobile application providers) offers its consumers information and communication services (e.g. multimedia clips, mobile applications) via Internet. Each service is characterized with its description (e.g. identification, category, author, tags) and data that defines the number of its consumers (e.g. multimedia clip viewers, mobile application downloads) from the moment it was introduced on the market.

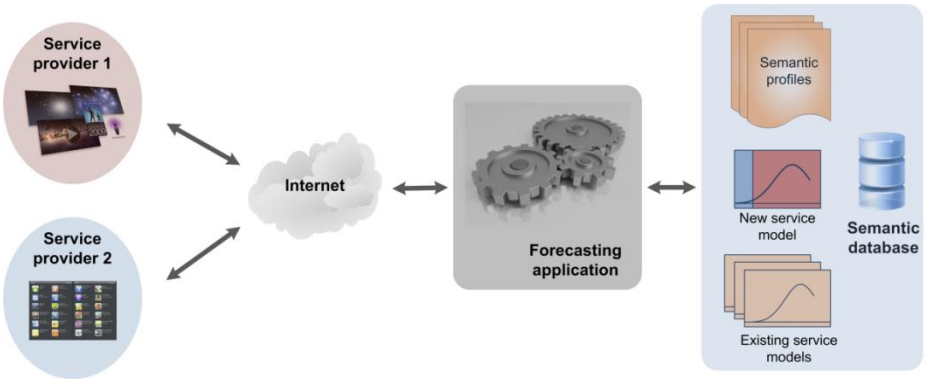

Fig. 6. Generic application architecture for predicting consumer interest in newly introduced services.

The *forecasting application* performs five main tasks:

i) collecting data about the services from the providers,
ii) transforming the service descriptions into structured semantic profiles later stored in a semantic database,
iii) identifying the appropriate growth model and its parameters from the service acceptance statistics,
iv) semantic service profile matchmaking, and
v) determining the prediction model and belonging parameters for a newly introduced service in the system.

The *semantic repository* is used for storing data the forecasting application collects and transforms into semantic service profiles. Service data collection

is commonly performed via APIs provided by the service providers (e.g. *YouTube API*[1], *Facebook Graph API*[2]). The data consists of service descriptions and statistical indicators of service acceptance (e.g. number of viewers or comments).

4.2. Forecasting system functionalities.

When it comes to service descriptions, most service providers maintain social tags that authors, and sometimes consumers, use to annotate the service. This folksonomy should then be translated into exact semantic concepts with unique identifiers within taxonomy, so that relations and similarities of these concepts can be rightfully detected when semantic profile matchmaking is being done [18][19][20].

Aside from service descriptions the system must also process the statistical service acceptance data. The forecasting application should use service acceptance data to identify the most appropriate growth model and calculate its parameters. The acceptance data for services that have been in the market for a considerable amount of time is most likely to exceed the necessary number of data points for calculating the model parameters. In such cases of overdetermined systems it is usually not possible to find the model that fits the data points exactly, so it is necessary to find the best possible approximation. Most common method used in such cases is called *least squared method*, as was mentioned earlier [17]. The task of choosing the model that suits the growth of each individual service best, comes down to finding the best approximation with each growth model taken into account and selecting the one which provides the minimal difference between the actual and modeled values.

Once the semantic service profiles are created, belonging models selected and their parameters calculated, all prerequisites are met for new service prediction model calculation. The algorithm for this task should solve two main problems: i) choosing the correct prediction model, and ii) calculating the parameters for the selected model. These problems should be observed taking into account the results of semantic comparison between new service's profile and the profiles of existing services. Services with greater similarities should have greater impact during model selection and parameter calculation.

5. Proof-of-concept scenario: modeling growth of a newly introduced YouTube clip.

This section will present the functionalities of our proposed system that were mentioned earlier. The proposed system implementation is shown in Fig. 7.

[1] YouTube API: *https://developers.google.com/youtube/*
[2] Facebook Graph API: *http://developers.facebook.com/docs/reference/api/*

We will use YouTube as a proof-of-concept service provider and multimedia clips as proof-of-concept information and communication services. YouTube clips are divided into categories (e.g. music, film & animation, education). Also, each clip has a short description suggested by the author. This information, along with technical characteristics (e.g. clip length and resolution) and acceptance data, is translated into a semantic profile as described in Sections 3 and 4.

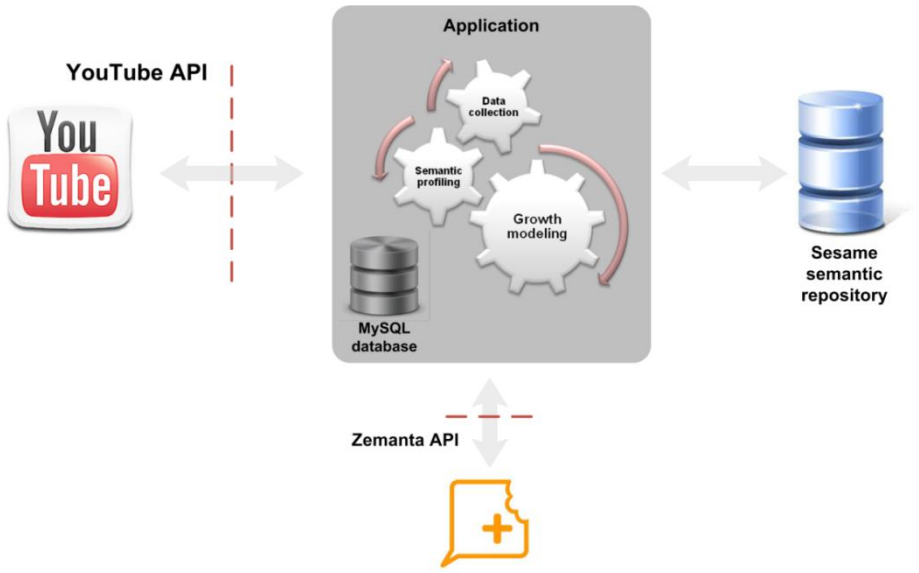

Fig. 7. Implementation of the proposed system for YouTube clip growth forecasting.

5.1. Data collection

The forecasting application is in constant communication with YouTube via the *YouTube API*. The API is used for collecting data on YouTube clips, such as the name and the author of the clip, the category, author's short description of the clip, clip duration, etc. Aside from this data, our forecasting application also requires clips' acceptance statistics and identification of the keywords that describe the clips' contents. This information cannot be collected via the API so it is necessary to use alternative resources.

The acceptance statistics are obtained by parsing of the XML retrieved from the *Insight API*[3]. The statistics include data on number of clip viewers, number of comments, likes and dislikes, and certain demographic data. It is necessary to mention that not all clips on YouTube have these statistics available.

[3] Insight API: *http://developers.facebook.com/docs/reference/api/insights/*

Initially YouTube also provided the tags that clip authors used to annotate their clips when uploading them. Due to increasing misuse of these tags in order to attract more interest to their clips, starting from August 2012 tags are no longer visible to viewers and cannot be obtained via the YouTube API. In order to extract the keywords for enabling semantic matchmaking of clip profiles, the forecasting application must take use of a text mining tool. Our application uses the *Zemanta API*[4] for processing short descriptions of the clips. For each text the API returns sets of entity URIs, related images, articles, hyperlinks, and tags for further use.

All obtained data is first stored in a relational database (i.e. *MySQL* database) so it can afterwards be used for semantic profiling and prediction model parameter calculation. The semantic annotation of each clip and growth model information will finally be stored as structured semantic profiles in a semantic repository.

5.2. Semantic profiling

The goal of semantic profiling in our case study is to recognize key concepts and measures that describe YouTube clips. This process assigns values to the three sets of attributes in the semantic service profile:

i) identification information,
ii) content information, and
iii) technical characteristics.

An example of semantic profiling is shown in Fig. 8.[5]

The *identification information* in our proof-of-concept scenario contains the following attributes and values:

- *is* – this attribute contains data regarding the type of the service. In our scenario all services will have the type *Video clip*. The URI of that type within the *DBpedia Ontology* is http://dbpedia.org/resource/Video_clip,
- *hasProvider* – names the service provider. In this case the service provider is YouTube (http://dbpedia.org/resource/YouTube),
- *hasID* and *hasURL* – define the unique service ID and the URL of the clip,
- *hasTitle* – declares the title of the video clip,
- *hasUploader* – identifies the username of the person who uploaded the video to YouTube.

The *content information* is divided into three sets of attributes:

i) category,
ii) keywords, and
iii) keyword URIs.

[4]Zemanta API: http://developer.zemanta.com/

[5]URL of the YouTube clip: *http://www.youtube.com/watch?v=8UVNT4wvIGY* (accessedinOctober 2012)

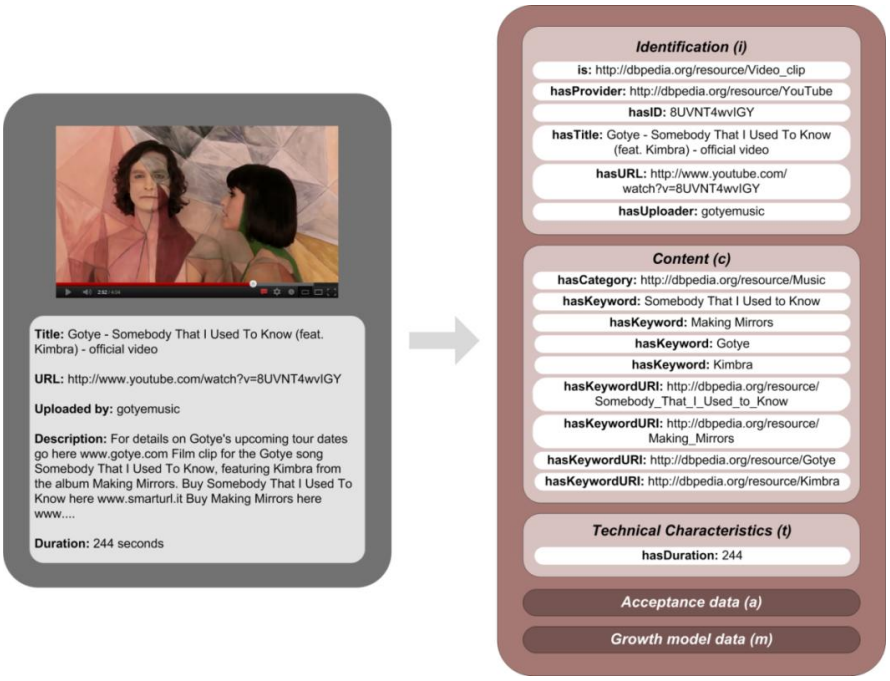

Fig. 8. Semantic profiling of a YouTube clip.

The attribute *hasCategory* identifies the category of the video clip. Our example shows a music video so the category is *Music* (http://dbpedia.org/resource/Music). Keywords obtained with the help of *Zemanta API* are assigned to attributes called *hasKeyword* and their belonging URIs that will be used for semantic matchmaking to attributes named *hasKeywordURI*. These attributes define key concepts for understanding and describing an information and communication service.

The only *technical characteristic* that can be obtained for a clip via the *YouTube API* is the duration of the clip expressed in seconds. This information is stored as value of the attribute *hasDuration*. The last two parts of the service profiles, the acceptance data and the growth model data, are obtained through growth modeling mechanisms which are described below.

5.3. Growth modeling

In order to perform growth forecasting for newly introduced services it is necessary to calculate the similarities between the new and the existing services, and determine the growth model and its parameters for each existing service. The similarities calculated by semantic matchmaking of service profiles represent the correlation between existing services' growth and diffusion of the new service in our case scenario. Similar YouTube clips

tend to attract the same viewer population, and due to the fact that the clip reproduction is virtually free for the viewers, there is no competitive influence among the services. Thus we can assume that similar clips should have similar growth.

Service matchmaking. Semantic matchmaking of service profiles must result in a decimal number between **0** and **1** which approximates the true semantic similarity of the two services. Even though various researches propose diverse metrics for expressing similarities between concepts we propose using values between **0** and **1** due to simplification of further use of these values. Matchmaking algorithm must include attributes from the identification part of the profile, the content describing attributes and technical characteristics. Each attribute must be observed separately because not all attributes have the same importance, and also the resulting similarity of each attribute depends on its meaning and type of value it can take on.

The attribute *is* reflects the type of the service, and as such has very high significance for the overall result. On the other hand, there is no point in comparing the values of attributes *hasID* and *hasURL* considering the fact these values are supposed to be unique. As was described in Section 2, comparing *string* values results in values 0 and 1 depending whether the values are a match or not, the result of numeric value matchmaking is the ratio of the two values, and finally resource matchmaking is performed with the use of *DBpediaOntology*. Table 1 shows an example of two service profiles that ought to be compared. These two clips are a music video by *Gotye* and *Kimbra*, already described in Fig. 8, and a summary of the soccer match between *Manchester United* and *Real Madrid* in the *UEFA Champions League* played in February 2013[6].

Semantic matchmaking is first performed between service types (i.e., attribute name = "*is*") and service providers (i.e., attribute name = "*hasProvider*"). In our scenario these values will be dbpedia.org/resource/Video_clipfor the service type and dbpedia.org/resource/YouTube for the service provider. Taking that into account, the semantic similarity for these attributes will be 1 for each pair of YouTube clips. Identification attributes *hasId* and *hasURL* will not be matched as they are unique for each clip. When it comes to attributes *hasTitle* and *hasUploader*, when they are a complete match the similarity is 1, otherwise it is 0. Another option for comparing string values is using more complex semantic similarity determination algorithms, though such approach could prove to be costly once the implementation is done in an environment with a great number of observed services [21][22]. We used the first, less complex metric for our scenario. In the case of the two clips shown in Table 1, the similarity regarding attributes *hasTitle* and *hasUploader* is 0 because the values are not a complete match.

[6] URL of the YouTube clip: *http://www.youtube.com/watch?v=PLpAMYubUeY* (accessed in February 2013)

Table 1. Service matchmaking example.

Att. group	Attribute name	Value type	Service No. 1	Service No. 2
Identification (i)	is	Resource	dbpedia.org/ resource/ Video_clip	dbpedia.org/ resource/ Video_clip
	hasProvider	Resource	dbpedia.org/ resource/YouTube	dbpedia.org/ resource/YouTube
	hasId	String	8UVNT4wvIGY	PLpAMYubUeY
	hasTitle	String	Gotye - SomebodyThat I Used To Know (feat. Kimbra) - official video	Real Madrid 1-1 Manchester United 2013 All Goals & Highlights CHAMPIONS LEAGUE 13-02-2013 HD
	hasURL	String	www.youtube.com/ watch?v= 8UVNT4wvIGY	www.youtube.com/ watch?v= PLpAMYubUeY
	hasUploader	String	gotyemusic	newsfootballnews
Content (c)	hasCategory	Resource	dbpedia.org/ resource/Music	dbpedia.org/ resource/Sport
	hasKeyword	String	Somebody That I Used to Know Making Mirrors Gotye Kimbra	Real Madrid C.F. Manchester United F.C. Goal Xavi FC Barcelona Andrés Iniesta Manchester derby Sergio Busquets Lionel Messi
	hasKeyword URI	Resource (dbpedia.org/ resource/)	Somebody_That_I_ Used_ to_Know Making_Mirrors Gotye Kimbra	Real_Madrid_C.F. Manchester_United_F. C. Goal_%28ice_hockey% 29 Xavi FC_Barcelona Andr%C3%A9s _Iniesta Manchester_derby Sergio_Busquets Lionel_Messi
Technical (t)	hasDuration	Number	244	188

When it comes to content part we observe attributes *hasCategory* and *hasKeywordURI*. Attribute *hasKeyword* is represented through the attribute *hasKeywordURI*, which contains the identifiers of each keyword within used ontology. These identifiers define the keywords so that computers can process them, thus providing the option of automated resource matchmaking taking into account the true meaning of these concepts. We perform resource matchmaking as was described in Section 2. Fig. 9 shows an example how two resources within ontology are compared. The similarity of the two resources *A* and *B* within ontology *O* can then be calculated using the algorithm presented in Listing 1.

```
1.     similarity (A,B) {
2.     if (A == B) then return 1;
3.     CCAB = {c ∈ classes(O)| c
4.     rdf:typeA,crdf:typeB};
5.     if (|CCAB| == {Ø}) then return 0;
6.     SSAB = {Ø};
7.     for each (c ∈ CCAB) {
8.     dA = distance(c,A);
9.     dB = distance(c,B);
10.    SSAB = SSAB + {2^(-max(dA,dB))};
11.    }
12.    return max(SSAB);
13.    }
14.    distance(c,A) {
15.        FLSC = {x ∈ classes(O)| x rdf:typeA,
16.    xrdfs:subClassOfc};
17.        DIST = {Ø};
18.        if (|FLSC|== 0) then return 1;
19.        for each (x ∈ FLSC){
20.              if (exists y ∈ FLSC – {x} |
21.    (xrdfs:subClassOfy)) then  FLSC = FLSC – {x};
22.        }
23.        for each (x ∈ FLSC){
24.              DIST = DIST + {distance(x,A)+1};
25.        }
26.        return min(DIST);
27.    }
```

Listing 1. Algorithm for calculating semantic similarity of two resources.

In our example from Fig. 9, we take resources *Gotye* and *Lionel Messi* as resources *A* and *B*, and *DBpedia* as ontology *O.* The set of common parent classes for the two resources, CC_{AB}, contains classes *Person*, *Agent* and *Thing*. If we observe class *Person* and its distance from resource *Gotye*, the first step is to populate the set of subclasses *FLSC* (first-level subclasses) with classes *Artist* and *MusicalArtist* (Listing 1, lines 15, 16). The next step is

to eliminate all classes from *FLSC* that are a subclass of another class within *FLSC* in order to eliminate all classes that are not direct (or first-level) subclasses of the class *Person* (Listing 1, lines 19-23). In this case the class *MusicalArstist* is a subclass of *Artist*, so the only remaining class in *FLSC* is *Artist.* If there was more than one path between the class from CC_{AB} (i.e., *Person*) and the resource (i.e., *Gotye*), *FLSC* would have more than one element and we would have to choose the shortest path between the class from CC_{AB} and the resource (Listing 1, line 27). That is the reason the algorithm returns the minimum value of distances between classes in *FLSC* and the resource.

In the next recursive iteration *FLSC* would contain the direct subclasses of *Artist* that the resource *Gotye* instantiates (i.e. *MusicArtist*). The last recursive iteration would return 1 because *MusicArtist* has no subclasses that *Gotye* instantiates, and that would be the distance between class *MusicArtist* and resource *Gotye*. The distance function for *Artist* and *Gotye* would return 2, and for *Person* and *Gotye* would return 3. That would finally be the value of d_A. The same algorithm is used to calculate the distance between *Person* and *Lionel Messi*, which would also be 3. If the distances were different it would be necessary to isolate the greater one as more relevant. The similarity between *Gotye* and *Lionel Messi* while observing the class *Person* would then be $2^{-max(3,3)} = 0.125$ (Listing 1, line 10). Using the same procedure, the set of similarities SS_{AB} would also be populated for classes *Agent* and *Thing* with values 0.0625 and 0.03125 respectively (Listing 1, line 10). Finally, similarity of the resources *Gotye* and *Lionel Messi* (Listing 1, line 12) would be the maximum of the three values (i.e. 0.125).

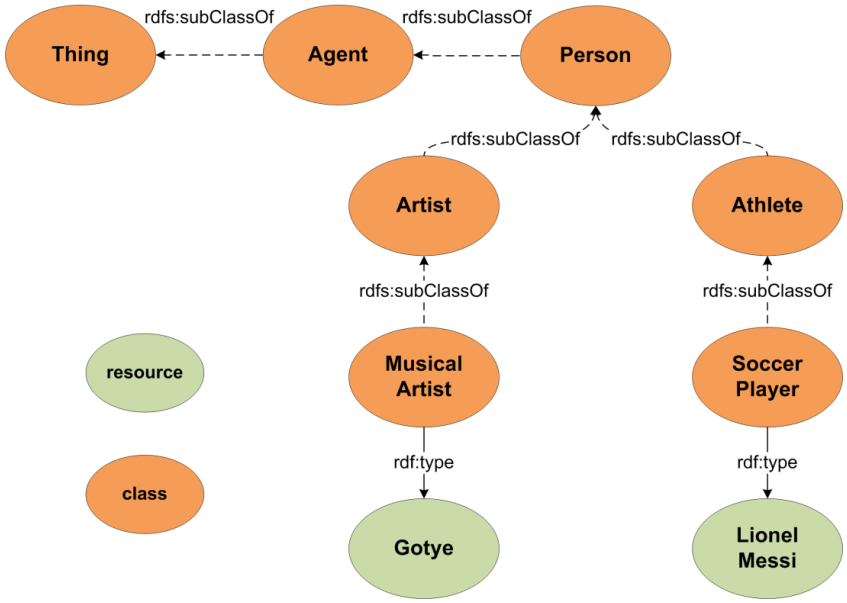

Fig. 9. Semantic matchmaking of two resources.

The final part used for service matchmaking, technical characteristics, is basically comparing the numeric values of the attribute *hasDuration*. The similarity between these two values is the quotient of the smaller and greater value. In the example from Table 1 the similarity would be 0.77 (i.e., $188/244$).

Once all attributes have been compared it is necessary to calculate the similarity between the profiles by using the values obtained by single attribute matchmaking. The function for calculating profile similarities should reflect the relevance of each attribute. We propose using the following mathematical model:

$$ss_{ij} = ss\left(p_{s_i}, p_{s_j}\right) = ss(i_{s_i}, i_{s_j})\left(w_c \times ss(c_{s_i}, c_{s_j}) + w_t \times ss(t_{s_i}, t_{s_j})\right), \quad w_c + w_t = 1 \tag{5}$$

where ss represents the semantic similarity between two sets of attributes, and w_c and w_t represent the relevance of the content information and technical characteristics respectively. Each attribute also has its weight factor that represents its relevance within the part of the profile it belongs to. For example, if we compare the identification and technical parts of the clips from Table 1, we get:

$$\begin{aligned} ss(i_1, i_2) &= w_{is} \times ss(is_1, is_2) + \\ &\quad w_{hasProvider} \times ss(hasProvider_1, hasProvider_2) + \\ &w_{hasID} \times ss(hasID_1, hasID_2) + \\ &w_{hasTitle} ss(hasTitle_1, hasTitle_2) + \\ &w_{hasURL} \times ss(hasURL_1, hasURL_2) + \\ &w_{hasUploader} \times ss(hasUploader_1, hasUploader_2) \\ &= 0.6 \cdot 1 + 0.2 \cdot 1 + 0 \cdot 0 + 0.1 \cdot 0 + 0 \cdot 0 + 0.1 \cdot 0 = 0.8 \end{aligned} \tag{6}$$

$$\begin{aligned} &ss(t_1, t_2) = w_{hasDuration} \times ss(hasDuration_1, hasDuration_2) \\ &= 1 \cdot 0.77 = 0.77 \end{aligned} \tag{7}$$

The comparison of content information is more complex than the other two parts because it is necessary to compare multiple resource values from both profiles thus generating a matrix of similarities for the attribute *hasKeywordURI*. If we were to use a mere arithmetic mean of the all the values from the matrix two clips would have a very low similarity even if they had identical sets of keywords. In order to override such problems we propose selecting a smaller number of highest values from the matrix. An example is shown in Table 2. We order the matrix so that the set of keywords with higher cardinality represents the rows in the matrix, and the other profile's keywords represent the columns. We then select the highest value from each row and include it in the calculation of content information similarity.

Table 2. Keyword concept similarity matrix.

	Somebody That I Used to Know	*Making Mirrors*	*Gotye*	*Kimbra*
Real Madrid C.F.	**0.03125**	0.03125	0.03125	0.03125
Manchester United F.C.	**0.03125**	0.03125	0.03125	0.03125
Goal (ice hockey)	**0.0625**	0.0625	0.03125	0.03125
Xavi	0.03125	0.03125	**0.125**	0.125
FC Barcelona	**0.03125**	0.03125	0.03125	0.03125
Andres Iniesta	0.03125	0.03125	**0.125**	0.125
Manchester derby	**0.00012**	0.00012	0.00012	0.00012
Sergio Busquets	0.03125	0.03125	**0.125**	0.125
Lionel Messi	0.03125	0.03125	**0.125**	0.125

We order the matrix so that the set of keywords with higher cardinality represents the rows in the matrix, and the other profile's keywords represent the columns. We then select the highest value from each row and include it in the calculation of content information similarity. The similarity for the attribute *hasKeywordURI* is the arithmetic mean of the selected values from the matrix. The content information similarity is then:

$$\begin{gathered} ss(c_1, c_2) = w_{hasCategory} \times ss(hasCategory_1, hasCategory_2) + \\ w_{hasKeywordURI} \times ss(hasKeywordURI_1, hasKeywordURI_2) \\ = 0.3 \cdot 0.5 + 0.7 \cdot 0.073 = 0.201 \end{gathered} \quad (8)$$

Having calculated similarities for each part of the profile we can calculate the profile similarities using (5). If we set the weight factors w_c and w_t to 0.75 and 0.25 respectively, we get the result 0.3433.

Modeling existing services. Growth modeling for existing services is done using the before mentioned *weighted least squares method*. The process of modeling must include calculating the model parameters for all models that are considered relevant in the system, and selecting the one that approximates the actual values best depending on the variance. For the purposes of our case study we used the logistic and Bass model. Fig. 10 shows growth modeling for the service presented in Fig. 8.

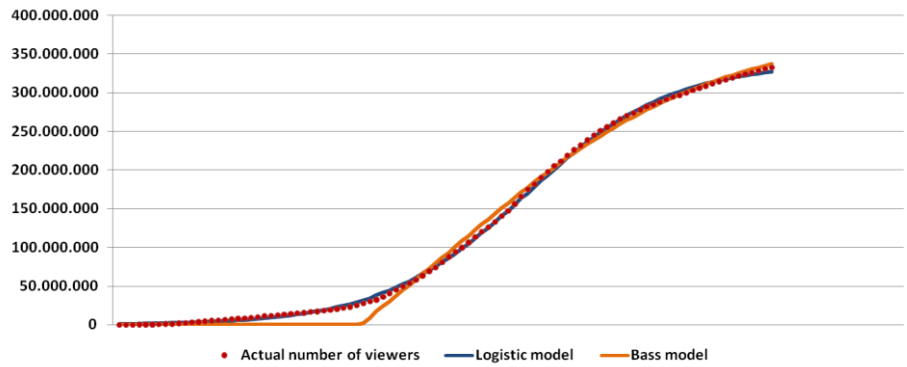

Fig. 10. Growth modeling for Service No.1 from Table 1.

The actual data points are approximated with two mentioned models using the weighted least squares method. There is exactly one hundred data points and each has its weight. The oldest one has weight 0.01, and the weight increases linearly until it reaches 1.00 at the most recent data point. Each model and its calculated parameters are evaluated by the calculated divergence from actual values. The divergence is calculated as the ratio of the weighted sum of squares of distances between actual and modeled values, and the sum of squared actual values. In our case the logistic model has proven to be more precise and as such will be considered more relevant but both will be stored within the service profile as model data m.

Table 3. Growth model parameters.

Model	**Logistic**	**Bass**
Parameters	M = 338 190 000 a = 7.087 b = April 19th, 2012	M = 379 000 000 p = 1.3 q = 2.7 t_s = December 21st, 2011
Divergence of modeled values (Δ)	0.016 %	0.125 %

Growth modeling for newly introduced services. When a new service is introduced into the market, the first step for determining its growth model and predicting its acceptance is to determine the set of most similar existing services. After the service has been compared to existing services and their similarities have been calculated, it is possible to detect a limited group of services that exceed a certain threshold of similarity, or simply select a certain number of services that are most similar to the one at hand.

Once the set of similar services has been determined we must first choose the model that would most likely approximate the new service's growth. The process of model selection must take into account the similarities between services and the divergences presented in Table 3. The divergence is inversed to the probability of selecting a prediction model for growth approximation. In our example from Table 3, the logistic model is far more likely to represent the future growth of the given YouTube clip.

Table 4 shows a set of similar services for the Service No. 2 from Table 1. The data shown in Table 4 contains information about service similarity and parameters and divergence of each model.

If the *Manchester United VS Real Madrid* clip (the Service No. 2 from Table 1) was just uploaded, a whole different modeling process would occur. The new video clip is compared to all existing clips in the system. After semantic matchmaking results are obtained we choose a certain number (e.g. *three*) of clips most similar to the newly introduced clip. The next step towards forecasting the clip's acceptance on the market would be selecting the right model. In order to do so, we used the semantic similarities and the divergences (Δ) calculated for each set of model parameters. The probability (ρ) of selecting model$n$ over others for growth description is calculated as follows:

$$\rho_n = \frac{\frac{1}{\Delta_n}}{\sum \frac{1}{\Delta_i}} \tag{9}$$

where i is the index that iterates through the set of similar services.

Knowing this, we can calculate the probability of for each model to be used for growth forecasting of the new service. We propose using the weighted arithmetic mean with semantic similarities being the weight factors:

$$\rho_n = \frac{\sum ss_{in} \times \rho_i}{\sum ss_{in}} \tag{10}$$

Table 4. Sample set of similar clips to the clip defined as the Service No. 2 from Table1.

YouTube clip ID[7]	ss	Logistic		Bass	
V7Kf8tGpb_w	0.468	M	185 000	M	176 000
		a	6.92	p	9.2
		b	May 10th, 2012	q	0.02
				t_s	May 2nd, 2012
		Δ (%)	0.04	Δ (%)	0.09
25PtkMLByLs	0.458	M	152 000	M	149 000
		a	65	p	75
		b	Jan 9th, 2013	q	10
				t_s	Jan 8th, 2013
		Δ (%)	0.25	Δ (%)	0.08
2LpRUHv9GQs	0.448	M	339 000	M	337 000
		a	19	p	30
		b	Aug 5th, 2012	q	0.2
				t_s	Aug 17th, 2012
		Δ (%)	0.03	Δ (%)	0.03
WrLQ-GT-O1s	0.317	M	81 000	M	81 000
		a	6.92	p	31.7
		b	May 10th, 2012	q	0.08
				t_s	Mar 13th, 2012
		Δ (%)	0.04	Δ (%)	0.03
OIhzrgk1wJ4	0.298	M	50 000	M	52 000
		a	13.4	p	0.1
		b	Sep 5th, 2012	q	14
				t_s	May 2nd, 2012
		Δ (%)	0.11	Δ (%)	0.24
…	…	…	…	…	…

[7]URL of each YouTube clip consists of the fixed part (*http://youtube.com/*) and the clip's ID.

In our case the probability of the new clip being modeled using the logistic model is 48%, and using the Bass model 52%. The logical choice would then be the Bass model. The next step would be calculating the model parameters. We suggest using the weighted geometric mean with the exception of parameter t_s that represents the point where the service is introduced and the growth starts. This parameter and the parameter b used in the logistic model are specific because they represent a point in time that cannot be calculated from parameters of other services. We suggest calculating the Δt in days between service introduction date and the point in time represented through these parameters. For example, the first service in Table 4 was introduced on May 26, 2012, so $b$would be presented by time difference of -16 days, and t_s by the time difference -24 days.Considering the day difference can be negative we suggest using the arithmetic mean instead. Finally, the parameters of the newly introduced clip are:

$$M_n = \left(\prod_{i=1}^{3} {M_i}^{ss_{in}}\right)^{1/\sum_{j=1}^{3} ss_{in}} = 211\,101.2 \tag{11}$$

$$p_n = \left(\prod_{i=1}^{3} {p_i}^{ss_{in}}\right)^{1/\sum_{j=1}^{3} ss_{in}} = 27.22 \tag{12}$$

$$q_n = \frac{\sum_{i=1}^{3} {q_i}^{ss_{in}}}{\sum_{j=1}^{3} ss_{in}} = 0.336 \tag{13}$$

$$t_{s_n} = \frac{\sum_{i=1}^{3} {t_{s_i}}^{ss_{in}}}{\sum_{j=1}^{3} ss_{in}} = -12.7\ days \tag{14}$$

where ss_{in} represents the semantic similarity between the new clip and YouTube clip i, while M_i, p_i, $q_i$and t_{s_i} are Bass model parameters of clip i. Once parameters M_i, p_i, $q_i$and t_{s_i} are calculated it is possible to approximate the number of viewers the new clip should reach in near future.

Having a larger number of clips for the scenario would increase the accuracy of forecasting. The accuracy would also benefit from increasing the number of iterations during model parameter calculations for existing services due to further minimization of divergence between actual and modeled values.

6. Conclusion and future work.

In this paper, we propose a semantic-aware model for forecasting consumer interest in new services. The innovativeness of our proposal can be recognized in using semantic reasoning for enhancing newly introduced service growth modeling. The semantic reasoning is particularly helpful when insufficient data about new service popularity is available – semantic reasoning enables us to substitute missing data for parameter calculation with the data from similar services already on the market. Such approach should enable service provider to perform pre-market forecasting in order to determine whether the service has its place in the market or it is destined for failure.

Our future research will be focused on the following four challenges. The first challenge is improving the implemented semantic reasoning mechanism. Key tasks which correspond to this challenge are improving scalability of semantic matchmaking algorithm in means of better utilization of different links between resources within *DBpedia Ontology*, and improving mechanisms for automated service profiling. The second challenge is implementing a more generalized model that will be applicable over complete service life-cycle (not just for the initial life-cycle phases). The third challenge would be optimizing the algorithm for implementing the weighted least squares method during existing service modeling. The final challenge is verification of our proposed system on various information and communication services (e.g. forecasting consumer interest in news based on news diffusion through a social network) using a real world benchmark dataset so the whole approach can be evaluated among common practices in service acceptance forecasting.

References

1. L. Vrdoljak, V. Podobnik and G. Jezic, Forecasting Consumer Interest in New Services Using Semantic-Aware Prediction Model: The Case of YouTube Clip Popularity, *Proc. of the 6th KES International Conference KES-AMSTA 2012,* Dubrovnik, Croatia, pp.454-463, 2012.
2. L. Vrdoljak, I. Bojic, V. Podobnik, G. Jezic and M. Kusek, Group-oriented Services: A Shift toward Consumer-Managed Relationship in Telecom Industry, *Transactions on Computational Collective Intelligence,* vol.2,pp.70-89, 2010.
3. D. Sreedhar, J. Manthan, P. Ajay, S.L. Virendra and N. Udupa, Customer Relationship Management and Customer Managed Relationship - Need of the hour, *http://www.pharmainfo.net/* (April 2011).
4. B. Schmitt, Customer experience management: a revolutionary approach to connecting with your customers, John Wiley and Sons, Hoboken, New Jersey, USA, 2003.
5. P.B. Girish, How Banks Use Customer Data to See the Future, *http://www.customerthink.com/* (April 2011).
6. N. Shin, Strategies for Generating E-Business Returns on Investment, Idea Group Inc, 2005.
7. C. Rygielski, J.C. Wang and D.C. Yen, Data mining techniques for customer relationship management, *Technology in Society*, vol.24, pp.483-502, 2002.
8. L. Vrdoljak, *Agent System based on Semantic Reasoning for Creating Social Networks of Telecommunication Service Users*, Diploma thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia, 2009.
9. M. Sokele, *Analytical Method for Forecasting of Telecommunications Service Life-Cycle Quantitative Factors*, Doctoral thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia, 2009.
10. M. Sokele and V. Hudek, Extensions of logistic growth model for the forecasting of product life cycle segments, *Advances in Doctoral Research in Management* (ed. L. Moutinho), World Scientific Publishing. vol.1, pp.77-106, 2006.
11. M. Sokele, Growth models for the forecasting of new product market adoption, *Telektronikk*, vol.4,no.3, pp.144-154, 2008.

12. F. Bass, A new product growth for model consumer durables, *Management Science*,vol.15, no.5, pp.215-227, 1969.
13. V. Mahajan, E. Muller and F. Bass, Diffusion of New Products: Empirical Generalizations and Managerial Uses, *Marketing Science*, vol.14, no.3, pp.79-88, 1995.
14. V. Mahajan, E. Muller and F. Bass, New Product Diffusion Models in Marketing: A Review and Directions for Research, *Journal of Marketing*, vol.54, pp.1-26, 1990.
15. Tellabs, Forecasting the Take-up of Mobile Broadband Services, White Paper, 2010.
16. U. Bellur, H. Vadodaria and A. Gupta, Semantic Matchmaking Algorithms, *Advances in Greedy Algorithms*, pp. 481-502, 2008.
17. NIST/SEMATECH e-Handbook of Statistical Methods, *http://www.itl.nist.gov/div898/handbook/* (October 2012).
18. Peters and W.G. Stock, "Power tags" in information retrieval, *Library Hi Tech*, vol.28, no.1, pp.81-93, 2010.
19. Zemanta – contextual intelligence for everyone, *http://developer.zemanta.com/* (October 2012).
20. Garcia-Silva, O. Corcho, H. Alani and A. Gomez-Perez, Review of the state of the art: Discovering and Associating Semantics to Tags in Folksonomies, *The Knowledge Engineering Review*, pp.1-24, 2004.
21. Y. Li, Z. Bandar and D. McLean, An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources, *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp.871-882, 2003.
22. Y. Li, D. McLean, Z. Bandar, J. O'Shea andK. Crockett, Sentence Similarity Based on Semantic Nets and Corpus Statistics, *IEEE Transactions on Knowledge and Data Engineering*, vol.18, no.8, pp.1138-1150, 2006.

Luka Vrdoljak is an Application Specialist and Test Designer at Intesa Sanpaolo Card Ltd., Croatia. He received his M.Sc. degree in electrical engineering from the University of Zagreb in 2009, where he is currently a PhD student. His research includes information services provisioning, customer relationship management and semantic web technologies.

Vedran Podobnik is an Assistant Professor at the Telecommunication Department of the Faculty of Electrical Engineering and Computing, University of Zagreb. He received M.Eng. (2006, Electrical Engineering) and Ph.D. (2010, Computer Science) degrees from the Faculty of Electrical Engineering and Computing, University of Zagreb. His research interests include social technologies, multi-agent systems, electronic markets, context-aware services and business process automation. He is a member of IEEE and KES International associations, as well as Cambridge Union Society.

Gordan Jezic is an Associate Professor at the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia. He received his B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Zagreb in 1995, 1999 and 2003, respectively. His research interests include communication networks and protocols, distributed computing, software agents and multi-agent systems and mobile process modeling. He is a member of IEEE, KES International and IEEE FIPA (The Foundation for Intelligent Physical Agents).

Received: September 20, 2012; Accepted: April 20, 2013