IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

Recommender System for Hotels based on user's personalized ratings using Hadoop and Cloud

J. S. Pawar[1], Nisha R.Patil[2], Harshala R.Khinde[3], Snehal S.Dandge[4], Sampada S. Bhavsar[5]

Assistant Professor, Department of Computer, PVG COE, Nashik, India[1]

Student, Department of Computer, PVG COE, Nashik, India[2,3,4,5]

Abstract: It is a service recommender system for providing appropriate recommendations to users. In the last decade, the number of customers, services as well as online information has grown rapidly. So, the big data analysis for service recommender systems is required. As a result, traditional service recommender systems often suffer from scalability and inefficiency problems when analysing such BigData. The vital thing is, most of existing service recommender systems present the same ratings of services to different users without considering previous user's preferences, and hence fails to meet user's personalized requirements. We will propose a method called "Keyword-Aware Service Recommendation", i.e. KASR, to fulfil the above challenges. In our system Keywords are used to indicate user's preferences. In very first stage data sets will be created for given system. Also user-based Collaborative Filtering algorithm is used to generate appropriate recommendations. KASR is implemented on Hadoop, to improve its scalability and efficiency in big data environment, a widely-used distributed computing platform known as MapReduce is used for parallel processing paradigm. At final stage, experiments are concluded on real-world data sets, and results shows that KASR significantly improves the accuracy & scalability of service recommender systems over existing approaches. As explained above we will use the techniques such as Map Reduce for parallel processing paradigm and the algorithm used in our system are Collaborative Filtering algorithm is for generating appropriate recommendations.

Keywords: KASR, ASC, ESC, Personalized rating, BigData, Cloud Computing, Hadoop.

I. INTRODUCTION

In day-to-day life there is large amount of data getting bigger from quite a while now. From less than a decade ago, mankind generated about 5 Exabyte of data. In 2012, global data grew upto 2.7 Zettabyte i.e., roughly 500 times more data than all data ever generated prior to 2003. Reason for data growing bigger is that, it is continuously being generated by variety sources such as sensors, CCTV cameras, social media and by variety devices. Much of that data such as Videos, Audios, Text Documents which means data is not stored on traditionally structured predefined tables. That's why the traditional data management and analytics tools alone do not enable IT to store, manage, process and also analyse BigData too.

A. *BigData*

"It refers to 'Data' whose size is beyond the ability of current technology to process, handle and capture the data within particular instance of time". Big Data likewise conveys new opportunities and distinguishing troubles to industry and the academia, like most BigData applications, the BigData tendency likewise postures overwhelming effects on service recommender techniques. With the developing number of options for services, effectively recommending services that users favoured have turn into an imperative research issue. Service recommender framework has been indicated as important tools to help users manage services over-burden and give proper recommendations to them. As data is so huge, it is very difficult to manage a data and it also takes a much more time to generate results. So this is a drawback of using BigData. But this issue can be dealt with the help of Hadoop, by using Hadoop a huge data can be analyzed in few seconds so Hadoop reduces the time.

B. *Hadoop:*

Hadoop is a distributed computing framework and released by Apache Foundation, it is Google's open source implementation of the cloud computing model, and also it can be efficient, reliable, scalable way to process data. Its core idea is to build on a large amount of cheap & efficient cluster hardware devices, in the form of software processing to pave the way of storage and computing environment for the huge amounts of data, and provide a unified standard interface, is a highly scalable distributed computing systems. While referred to HDFS distributed fie systems, to improve fault tolerance in the form of software. When we compared with the traditional file system, it has low cost, easy to expand, and high fault tolerance features. MapReduce is provided by Hadoop parallel computing model for handling large amounts of data calculation. Hadoop is scalable, it can easily meet the requirements of large-scale data need to handle on the PB-Level. In addition, the use of relatively low cost Hadoop cluster nodes require low internal computer, on any inexpensive computer can deploy In order to combined Hadoop with practical application better, you can also use some subprojects on the basis of Hadoop, such as MapReduce technology, HBase distributed data storage systems, scalable data warehouse Hive, high-level data flow language Pig, high-performance distributed

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

collaborative services ZooKeeper and other major use MapReduce technology and HBase data storage system to solve the problem of information retrieval.

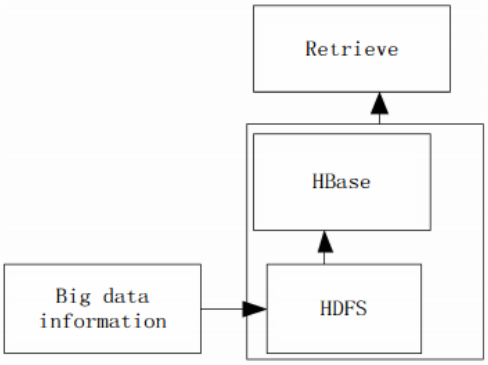

Fig. Hadoop Framework

Hadoop is a cluster computing system which is data intensive. In this system incoming jobs are developed using the MapReduce programming model.

II. PROBLEM DEFINITION

In traditional service recommender systems we might be suffer from scalability and inefficiency problems, while processing or analysing large-scale data. They haven't considered user's different preferences, and failed to meet user's personalized requirements. In this system, we propose a Keyword-Aware Service Recommendation method, named KASR, to achieve an efficiency of a Recommender System. It aims at presenting a personalized service recommendation list and recommending the most appropriate services to the users effectively.

III.LITERATURE SURVEY

In the mid-1990s recommender system emerged as an independent research area and when researchers started focusing on recommendation problems that rely on the ratings structure. In its most common formulation, the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a user [2]. Gediminas Adomavicius, and Alexander Tuzhilin gave an overview of the field of recommender systems and also described their limitations and gave the probable extensions of the existing recommender system of those times in their paper toward the next generation of recommender systems : a survey of the state-of-art and possible extensions, in 2005. Recommender systems are usually classified into the below categories, based on how recommendation are made.

A. Content-based recommendations:
User will be recommended items with ones the user preferred in the past.

B. *Collaborative recommendations:*
User will be recommended items that people which having similar tastes and the preferences liked in the past.

C. *Hybrid approaches:*
In hybrid approaches combination of collaborative and content-based methods is done. [2] They used utility functions, heuristic approaches to find the similarities and obtain recommendations. But many industries have begun employing multi-criteria systems. Hence they were not sufficient. Therefore, The author Gediminas Adomavicius and Young Ok Kwon proposed multi-criteria recommender systems in the paper named as "New recommendation techniques for multi criteria rating systems." Multi-criteria recommender systems are used to find items that are most useful to each user, just as single-rating recommender systems do. By that overall analysis we can say that the system must be able to predict each item's overall ratio for each user. So, that it can compare the items on the basis of its overall ratings and recommend the best items to the users. The difference between single rating and multi criteria rating systems is that the latter have more information about the users and items to use in the recommendation process [3] However, these multi-criteria problems typically do not apply to personalization and recommendation settings. Replacing that they involve finding solutions or items that are optimal in general (that is, optimal with respect to all users) and it do not explicitly consider differences in individual user preferences.[3]

So to overcome above addressed problems there came user profiling approaches, Huizhi Liang, YueXu, Jim Hogan proposed "Parallel User profiling based on Folksonomy for Large Scale Recommender System." An implementation of Cascading MapReduce, in 2010, User profile is used to describe user's interests and preferences information. Generally, an explicit or implicit rating vector is used in collaborative filtering based recommender systems to profile a user's preferences or interest to the items. With Folksonomy information, we can use a set of tags with their correspondent weights to profile users' topic interests [4]. They used k- nearest neighbour algorithm, and pipeline approaches. But this system is very costly and user dependant user has to login every time so was not much preferable.

Then Bayesian network found its roots in recommendations, Xiwang Yang, Yang Guo and Yong Liu propose a Bayesian-inference based recommendation system for social networks. In this system, users share their content ratings with friends. A user propagates a content rating query along the social network to his direct and indirect friends. Based on the query responses, a Bayesian network is built up to infer the rating of the querying user. They showed that Bayesian-inference based recommendation was more accurate than the traditional Collaborative Filtering (CF) recommendation and the existing trust-based recommendations. They used distributed protocol & also used Prior distribution to cope with cold start and rating sparseness. But it in the era of Big Data it was very difficult to cope up with users different choices and requirements so as to provide user with their personalized requirements we propose KASR a keyword aware service recommendation system based on Hadoop and cloud computing.

Comparing with existing methods, KASR utilizes reviews of previous users to get both of user preferences and the quality of multiple criteria of candidate services, which

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

makes recommendations more accurate. Moreover, KASR on MapReduce has favourable scalability and efficiency.[1]

IV. PROPOSED RECOMMENDATION METHOD

In this paper we refer the personalized recommendation system in which Keyword-candidate List and Domain Thesaurus are maintained for particular system. Preferences will be given by previous users. And then similar users are searched out by keyword extraction method also by calculating similarity. After that the keywords are classified and weights of reviews of similar users are calculated. Finally, recommendation list of top-k items is generated by the system.

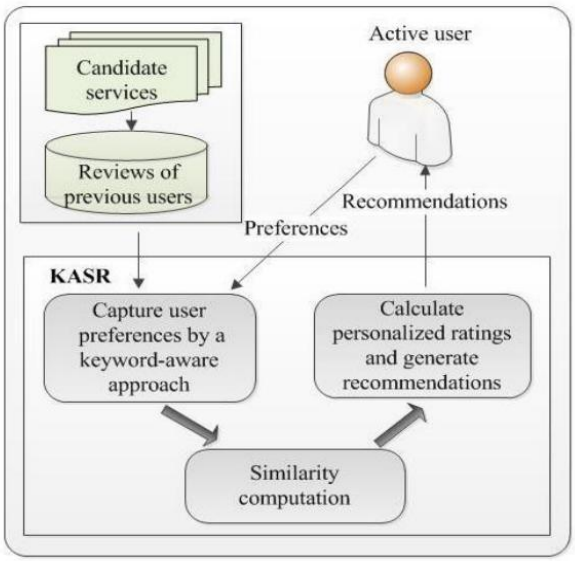

Fig. Working of System

In proposed system, keywords are used to indicate both i.e. users' preferences and the quality of candidate services. A user based CF algorithm is adopted to generate perfect recommendations. Proposed system which we are developing for calculating a personalized rating of each candidate service for a user, and then presenting a personalized service recommendation list.

Table1 summarizes the basic symbols and notations used in next algorithms.

TABLE I: BASIC SYMBOLS AND NOTATIONS

Symbol	*Definition*
K	The Keyword Candidate List, K={K1,K2..Kn}
APK	The Preference Keyword Set of active user
PPK	The Preference Keyword Set of previous user
Sim(APK,PPK)	The similarity between APK and PPK
	Preference Weight Vector
	Preference weight vector of active user
	Preference Weight vector of previous user

In our system, two data structures as “keyword candidate list” and “specialized domain thesaurus”, are introduced to obtain users' previous users preferences, which is shown in next table.

In our system, following table shows the concept of domain thesaurus by using “keyword candidate list” and “Specialized domain thesaurus” introduced to obtain users' previous users preferences.

A. *Keyword Candidate List and Domain Thesaurus:*

The keyword candidate list consists of set of keywords about previous user’s preferences and multi-criteria of the candidate services, which can be denoted as K= 1.where, n is the number of the keywords in the keyword candidate list. An example of a simple keyword candidate list of the hotel reservation system is described in Table 2.

No.	***Keyword***	***No.***	***Keyword***
1	Shopping	2	Food
3	Room	4	Value
5	Fitness	6	Environment
7	Service	8	Transportation

Keyword candidate list consist of no of keywords. Keywords in the keyword candidate list can be a word or multiple words related with the particular quality criteria of candidate services. In this system, the preferences of previous users will be extracted from their reviews or comments for candidate services and formalized into a keyword set. Since, some of words in reviews cannot exactly match the corresponding keywords in the keyword candidate list which characterize the same aspects as the words. The related keywords should be extracted as well. In KASR, specialized domain thesauruses are built. It is used to support the keyword extraction for calculating ratings. And also the different domain thesauruses are built for different service domains.

Domain thesaurus: A domain thesaurus is a set of keywords which consist of the keyword candidate list which lists words grouped together according to the similarity of keyword synonyms including related and contrasting words as well as antonyms.

B .User Preferences:

In this step, the previous preferences of active users and previous users are formalized into their particular preference keyword sets respectively. In this system, an active user refers to a current user needs recommendation. Preferences of an active user: An active user can give his/her preferences about candidate services by selecting keywords from a keyword candidate list, which reflect the quality criteria of the services he/she is concerned about. The preference keyword set of the active user can be denoted as APK= {ak1, ak2, ...akl} where aki ($1 \leq i \leq l$) is the ith keyword selected from the keyword candidate list by the active user, l is the number of selected keywords.

Preferences of previous users:

The preferences of a previous user for a candidate service are extracted from his/her reviews for the service according to the keyword candidate list and domain thesaurus. And a review of the previous user will be formalized into the preference keyword set of him/her, which can be denoted as PPK= {pk1, pk2,...pkh} where pki ($1 \leq i \leq h$) is the ith keyword extracted from the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

review, h is the number of extracted keywords. The keyword extraction process is described as follows:

Pre-processing:
Firstly, HTML tags and stop words in the reviews snippet collection should be removed to avoid affecting the quality of the keyword extraction in the next stage. And the Porter Stemmer algorithm (keyword stripping) is used to remove the commoner morphological and in flexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems.

Keyword Extraction:
In this phase, each review will be transformed into a corresponding keyword set according to the keyword candidate list and domain thesaurus. If the review contains a word in the domain thesaurus, then the corresponding keyword should be extracted into the preference keyword set of the user. For example, if a review of a previous user for a hotel has the word “spa”, which is corresponding to the keyword ``Fitness'' in the domain thesaurus, then the keyword ``Fitness'' should be contained in the preference keyword set of the previous user. If a keyword appears more than once in a review, the times of repetitions will be recorded. In this method, it is regarded that keywords appearing multiple times are more important. The times of repetitions will be used to calculate the weight of the keyword in preference keyword set in the next step.

Classify Keywords:
In this phase, each review will be transformed into a corresponding keyword set according to the keyword candidate list and domain thesaurus. If the review contains a word in the domain thesaurus, then the corresponding keyword should be extracted into the preference keyword set of the user. For example, if a review of a previous user for a hotel has the word “spa”, which is corresponding to the keyword ``Fitness'' in the domain thesaurus, then the keyword ``Fitness'' should be contained in the preference keyword set of the previous user. If a keyword appears more than once in a review, the times of repetitions will be recorded. Also in our system, it is regarded that when keywords are appearing multiple times are more important. The times of repetitions will be used to calculate the weight vector of the keyword in preference keyword set in the next step.

Similarity Calculation:
In the second step, we identify the reviews of previous users who have similar tastes to an active user. Before similarity computation, the reviews of previous user are unrelated to the active user's preferences will be filtered out. And this filtered by using the set theory of intersection in mathematics. If the intersection of the preference keyword sets of the active user and a previous user is an empty, then the preference keyword set of the previous user will be filtered out. Otherwise, two similarity computation methods are introduced in our system one as Approximate Similarity computation and second as Exact similarity computation. The approximate similarity computation is applicable when the weights of the keyword in the preference keyword set are not present in the domain thesaurus, while the exact similarity computation is for the case that the weight of the keywords are available.

A. *Approximate Similarity Computation:*
A frequently used method for comparing the similarity, Jaccard coefficient, is applied in the approximate similarity computation. The most important thing is Jaccard index i.e. Jaccard coefficient is measurement of asymmetric information on binary (and no binary) variables, and it is useful when negative values give no information. The similarity between the preferences of the active user and a previous user based on Jaccard coefficient is given as follows:

Sim(APK,PPK)=Jaccard(APK,PPK)=
|Intersection (APK,PPK)|/Union(APK,PPK)

Where APK is stands for preference keyword set of the active user, PPK is stands for preference keyword set of a previous user. And the weight of the keywords is calculated by using the concept weight vector.

Algorithm1: SIM-ASC(Approximate Similarity Computation)

Input: The Preference keyword set of active user APK. The preference keyword set of previous user PPK.

Output: Similarity of APK and PPK i.e. Sim(APK,PPK)
1.Sim(APK,PPK)=Jaccard(APK,PPK)=
|Intersection(APK,PPK)|/Union(APK,PPK)
2. Return similarity of APK and PPK.

B. *Exact Similarity Computation:*
A cosine-based approach is applied in the exact similarity computation, which is similar to the Vector Space Model (VSM) in information retrieval [24] [25]. **Definition 4 (Preference weight vector).** In this cosinebased approach, The preference keyword sets of the active user and previous users will be transformed into *n* dimensional weight vectors respectively, namely preference weight vector, which can be denoted as *WP* [*w*1,*w*2,...,*wn*], *n* is the number of keywords in the keyword-candidate list, *wi* is the weight of the keyword.
Firstly, we construct the pair-wise comparison matrix in terms of the relative importance between each two keywords. The pair-wise comparison matrix *A*m = (*aij*) *m* must satisfy the following properties, *aij* represents the relative importance of two keywords:

aij = 1 i=j=1,2,3,...m.
aij = 1/aji i=j=1,2,...m and i =! j
aij = aik/ajk i=j=k=1,2,...m and i =! j

Therefore, Weight vector will be given as,

$$w_i = \frac{1}{m}\sum_{j=1}^{m}\frac{a_{ij}}{\sum_{k=1}^{m} a_{kj}} \quad \text{....(1)}$$

Where, aij = relative importance between two keywords.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

m = number of the keywords in the preference keyword set of the active user.

The weight vector of the preference keyword set of a previous user can be decided by the 0TF-IDF stands for term frequency/inverse document frequency measure [4], which is one of the best-known measures for specifying the weight of keywords in Information Retrieval. Term Frequency, term frequency of the keyword pkiin the preference keyword set of user *u*'is defined as,

$$TF = \frac{N_{pk_i}}{\sum_g N_{pk_i}} \quad ...(2)$$

Where, Npki = the number of occurrences of the keyword pkiin all the keyword sets of the reviews commented by the same previous user.
g = the number of the keywords in the preference keyword set of the user.

Similarly, the term IDF is defined as,

$$IDF = \log \frac{|R'|}{|r': pk_i \in r'|} \quad ...(3)$$

Where, |R'| = the total number of the reviews commented by previous user, and |r': pki∈r'| is the number of reviews where, keyword pkiappears.

Hence weight of keyword can be calculated as,

$$w_{pk_i} = TF \times IDF = \frac{N_{pk_i}}{\sum_g N_{pk_i}} \times \log \frac{|R'|}{|r': pk_i \in r'|} \quad ...(4)$$

Similarly, cosine approach will be given as,

$$sim(APK, PPK) = \cos(\vec{W}_{AP}, \vec{W}_{PP}) = \frac{\vec{W}_{AP} \bullet \vec{W}_{PP}}{\|\vec{W}_{AP}\|_2 \times \|\vec{W}_{PP}\|_2}$$

$$= \frac{\sum_{i=1}^{n} \vec{W}_{AP,i} \times \vec{W}_{PP,i}}{\sqrt{\sum_{i=1}^{n} \vec{W}^2_{AP,i}} \sqrt{\sum_{i=1}^{n} \vec{W}^2_{PP,i}}} \quad ...(5)$$

Algorithm1:SIM-ESC(Exact Similarity Computation
Input: The preference keyword set of the active user APK The preference keyword set of a previous user PPKj

Output: The similarity of APK and PPKj,simESC(APK,PPK j)

1: for each keyword ki in the KCL
2: **if** ki APKthen
3: get Weight of active user preference calculated by formula (1).
4: **else** it is 0
5: end if
6: if ki PPKjthen
7: get weight of previous user preference calculated by formula (4).
8: else it is 0
9: end if
10: end for
11: get simESC(APK,UPK j) by formula (5)
12: return the similarity of APK and PPKj, simESC (APK,UPK j)

C. *Calculate Personalise ratings and generate recommendation:*
Filtering will be calculating by using similarity of active user and passive user which given the value as ∂. It is known as threshold value. Given a threshold , if *sim*(*APK*,*PPKj*) , the preference keyword set of a previous user PPKj will be filtered out, otherwise PPKj will be retained.

V. EXPERIMENTAL EVALUATION

By experiment we can verify that the scalability of KASR, in a cluster of nodes ranging from 1 to 8 respectively. There are four synthetic datasets used in the experiment they are as 128M, 256M, 512M and 1G data size. The following figure shows the speedup of KASR with respect to number of nodes .And the larger dataset obtained a better speedup. When the data size is 1G and the number of nodes is 8, the speedup value reaches 6.412, which is 80.15% (6.412/8=80.15%) of the ideal speedup. The experimental result shows that KASR on Map-Reduce in Hadoop platform has good scalability over "Big Data" and performs better with larger dataset.

These experimental results show that, KASR performs well in accuracy, and KASR on Map-reduce framework has good scalability in "Big Data" environment.

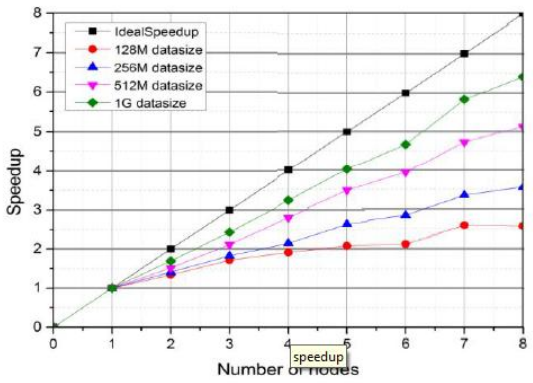

Fig. Speedup of KASR

VI. CONCLUSION

In this system, we have proposed a keyword-aware service recommendation method, named KASR. In KASR, keywords are used to indicate users' preferences, and a user based Collaborative Filtering algorithm is chosen to generate appropriate recommendations. For effective working, a keyword-candidate list and domain thesaurus are provided to help obtain users' preferences. The active user gives their preferences by selecting the keywords from the keyword-candidate list, and the preferences of the previous users can be taken out from their reviews for services according to the keyword-candidate list and domain thesaurus. Our system focuses on presenting a personalized service recommendation list and recommending the most appropriate services to the users.

DOI 10.17148/IJARCCE.2016.5221

Moreover, to improve the scalability and efficiency of KASR in "BigData" environment, we have implemented it on a MapReduce framework in Hadoop platform. Finally, the experimental results demonstrate that KASR greatly improves the accuracy and scalability of service recommender systems over existing approaches.

ACKNOWLEDGEMENT

Every work is source which requires support from many people and areas. It gives us proud privilege to complete the Project on "Recommender System for Hotels based on user's personalized ratings Using Hadoop & Cloud" under valuable guidance and encouragement of my guide **Prof. J. S. Pawar** and H.O.D(Computer Dept.) **Prof. M. T. Jagtap** and for providing all facilities and every help for smooth progress of project. We would also like to thank all the Staff Member of Computer Engineering Department for timely help and inspiration for completion of the project. At last we would like to thank all the unseen authors of various articles on the Internet, helping me become aware of the research currently ongoing in this field and all my colleagues for providing help and support in our work.

REFERENCES

[1] Shunmei Meng, Wanchun Dou, Xuyun Zhang, Jinjun Chen," KASR: A Keyword-Aware Service Recommendation Method on MapReduce for Big Data Applications", IEEE 2013.
[2] Gediminas Adomavicius, Member, IEEE, and Alexander Tuzhilin," Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE June 2005.
[3] Huizhi Liang, Jim Hogan, Yue Xu, "Parallel User profiling based on folksonomy for Large Scaled Recommender Systems". IEEE 2010.
[4] G. Salton, "Automatic Text Processing," Addison-Wesley, 1989.

DOI 10.17148/IJARCCE.2016.5221