

تشخیص وزن عروضی اشعار فارسی با استفاده از زنجیره مارکوف

محمد مهدی مجیری^۱؛ امیرحسین مجیری^۲

چکیده

یکی از مشکل ترین بخش های رشته ی ادبیات فارسی تشخیص وزن عروضی است، به صورتی که تشخیص وزن عروضی در بعضی اشعار برای اساتید ادبیات هم مشکل است. با توجه به اینکه تاکنون پژوهش چندانی در این زمینه انجام نشده، ورود به این عرصه ضروری به نظر می رسد. در ابتدای این مقاله در مورد پژوهش قبلی در زمینه تشخیص وزن عروضی اشعار فارسی صحبت کرده و با استفاده از الگوریتم پژوهش قبلی یک بیت شعر فارسی را به رشته ای از صفر و یک تبدیل می نماییم. با تبدیل هشت الگوی وزن عروضی اشعار فارسی به اتوماتا، با استفاده از زنجیره مارکوف وزن بیت ورودی را تشخیص می دهیم. نتایج این تحقیق بهبود در برخی اوزان را نسبت به پژوهش قبلی نشان می دهد.

کلمات کلیدی

وزن عروضی اشعار فارسی، ادبیات، زنجیره مارکوف، تشخیص الگو

Persian Poem Rhythm Recognition by Markov Chain

Mohammad Mahdi Mojiry; Amir Hossein Mojiry

ABSTRACT

The poem rhythm recognition is one of the most difficult parts of Persian literature. Since there is a few previous works in this realm, it is an interesting arena for research. In this article, we first discover the previous work on this topic. We convert a versa text to 0-1 string by last method. The system recognize the rhythm by use the Markov chain. In this method, automata design for eight of rhythm. The result show better accuracy in some of rhythm.

KEYWORDS

Persian Poem Rhythm, Automatic Rhythm detection, Markov Chain, Pattern Recognition

^۱ دانشجوی کارشناسی ارشد دانشکده کامپیوتر، دانشگاه آزاد اسلامی واحد تهران مرکز Moh.Mojiri.eng@iauctb.ac.ir

^۲ دانشجوی کارشناسی ارشد زبان شناسی، دانشگاه علامه طباطبائی amojiry@gmail.com

ادبیات فارسی، زمینی بکر است که به جهات مختلف، کسی جز در موارد نادر حاضر به ورود به این عرصه و زدن حرف نو نیست. برخی بر این باورند که آنچه گذشتگان کرده‌اند فصل الخطاب کاری است که می‌توان کرد و برخی از آن می‌ترسند که مورد طعن دیگران قرار گیرند. به عنوان مثال قبل از رایج شدن عروض علمی، عروض فنی غریب بود که کمتر کسی از اهل ادب با معیارهای آن آشنایی داشت و لذا آنان که عروض می‌دانستند در هاله‌ای از اسرار به نظر می‌رسیدند. [1]

هر چه قدر حرف نو زدن در ادبیات مشکل است، ورود کامپیوتر به این عرصه از آن مشکل‌تر. با وجود پیشرفت‌های فراوان در تحقیقات کامپیوتری، (به صورتی که محققان در برخی زبان‌ها همچون انگلیسی و اسپانیایی موفق به سرایش شعر به وسیله کامپیوتر شده‌اند [2]، [3]، [4]، [5] و [6]) هنوز بسیاری از ادیبان ما حاضر به قبول توانایی‌های این ابزار نیستند.

یکی مشکل‌ترین بخش‌های ادبیات فارسی تشخیص وزن عروضی اشعار است. تشخیص خودکار وزن عروضی توسط کامپیوتر برای اولین بار با استفاده از روش KNN در [7] انجام شد. این پژوهش تلاشی دوباره با استفاده از زنجیره مارکوف برای تشخیص وزن عروضی است. نتایج به دست آمده بهبود در برخی اوزان و ضعف در برخی دیگر را نشان می‌دهد.

۲. کارهای پیشین

تنها کار ثبت شده در زمینه تشخیص خودکار وزن عروضی اشعار فارسی [7] می‌باشد. با توجه به تفاوت فاحشی که بین وزن عروضی در زبان‌های مختلف وجود دارد، کارهای مشابه در زبان‌های دیگر قابل طرح در این مقوله نمی‌باشد. همچنین یافتن وزن در برخی زبان‌ها اهمیت چندانی ندارد.

این اولین بار نیست که تشخیص الگو با استفاده از زنجیره مارکوف انجام می‌شود. در [8] و [9] و [10] با استفاده از زنجیره مارکوف فرآیند تشخیص الگو انجام شده است. ما نیز با الگویی از این منابع اقدام به تشخیص الگوی وزن عروضی اشعار فارسی نموده‌ایم.

۳. وزن عروضی

وزن، نظم و تناسب خاصی است در اصوات شعر (=هجاءها) این نظم و تناسب اصوات به انحای گوناگون نزد ملل مختلف مبین نوعی آهنگ و موسیقی است. در شعر سنتی هر زبانی، تساوی هجاهای هر مصراع، در وزن دخیل است. علاوه بر این عامل مشترک، وزن شعر هر زبانی مبتنی بر عامل خاصی است، به عنوان مثال وزن اشعار انگلیسی و آلمانی مبتنی است بر تکیه که بر هجاهای واقع می‌شود. شعر فارسی به طور کلی مبتنی بر کمیت (کوتاه و بلندی) هجاءهاست. [۱] عروض برای شناختن اجزاء شعر و تشخیص وزن صحیح از وزن شکسته و ناصحیح است و یادگیری آن هرگز برای به دست آوردن توانایی لازم برای سرودن اشعار نیست. [۸]

۴. الگوریتم مورد استفاده

در [7] یک الگوریتم شش مرحله‌ای برای به دست آوردن وزن عروضی پیشنهاد شده است. در مرحله پنجم این الگوریتم، بیت ورودی به یک رشته از «U» و «-» تبدیل می‌شود. برای سادگی رشته مذکور را رشته‌ای از سفر و یک می‌نامیم.

در واقع با رسیدن به مرحله پنجم رشته‌ای از صفر و یک را داشته و تنها نیاز به انطباق آن به یکی از الگوهای وزن عروضی داریم. در پژوهش قبلی برای مرحله ششم از روش KNN استفاده شده است.

ما در این مقاله از پنج مرحله اول الگوریتم ابداع شده در [7] استفاده و رشته‌ای از صفر و یک را استخراج کرده‌ایم سپس با استفاده از زنجیره مارکوف، وزن بیت ورودی را تشخیص می‌دهیم.

۵. مجموعه دادگان

برای سنجش نتایج این تحقیق با پژوهش قبلی، از ویرایش دوم مجموعه دادگان [7] قابل دریافت از [9] استفاده کرده‌ایم. البته برای آنکه گراف تشکیل شده بسیار بزرگ شده و محاسبات آن پیچیده می‌شد، تنها از هشت وزن مشهورتر استفاده کردیم. بنابراین مجموعه داده مورد استفاده

شامل حدود ۸۰۰ بیت شعر است. هشت وزن مورد استفاده در **Error! Reference source not found.** آورده شده است. به هر یک از کلمات وزن رکن می‌گوییم. مثلاً «فعلاتن» یک رکن است و رشته معادل آن «-UU» با طول ۴ است.

جدول (۱) هشت وزن مورد استفاده

وزن	معادل - لارشته	کد
فعلاتن مفاعیلن فععلن	UU--U-U-UU--	28
فعلاتن فعلاتن فعلاتن فععلن	UU--UU--UU--UU--	29
مفاعیلن فعلاتن مفاعیلن فععلن	U-U-UU--U-U-UU--	30
مفعول فاعلات مفاعیل فاععلن	--U-U-UU--U-U--	31
مفعول مفاعیل مفاعیل فععلن	--UU--UU--UU--	32
فاعلاتن فاعلاتن فاعلاتن فاععلن	-U--U--U--U--	33
مفاعیلن مفاعیلن مفاعیلن مفاعیل	U--U--U--U--U	34
مفاعیلن مفاعیلن فععلن	U--U--U--	35

۶. تشخیص الگو با استفاده از زنجیره مارکوف

در [10] و [11] و [12] با استفاده از زنجیره مارکوف فرآیند تشخیص الگو انجام شده است. ما نیز در این پژوهش با استفاده از زنجیره مارکوف سعی در تشخیص الگوهای وزن عروضی اشعار فارسی می‌نماییم.

۵.۱. تشکیل ماتریس احتمال گذار

برای تشکیل گراف اولیه و درواقع برای به دست آوردن ماتریس احتمال گذار ابتدا گرافی از الگوهای هشت‌گانه وزن بر اساس ارکان موجود در آنها رسم می‌کنیم. این گراف در شکل (۱) آورده شده است. این گراف در واقع اتوماتای تولید این هشت وزن است. در مرحله‌ی بعدی نودهای گراف را عدد گذاری کرده تا بتوان به راحتی تبدیل به ماتریس گذار شود. این مورد در شکل (۲) نمایش داده شده است. پس از به دست آمدن گراف باید آن را وزن دار کرد. وزن نهایی یال‌های گراف از ترکیب وزن اولیه و وزن پویا به دست می‌آید. نحوه به دست آوردن این دو وزن و ترکیب آنها در بخش‌های بعدی گفته خواهد شد. در مرحله نهایی با وجود یک گراف وزن دار، ماتریس احتمال گذار تشکیل شده است.

۶.۲. وزن اولیه گراف

وزن اولیه یالها بر اساس تعداد تکرار آنها در هشت وزن به دست آمده است. به عنوان مثال در جایگاه اول دو وزن رکن «فعلاتن» آمده است ولی تنها در جایگاه اول یک وزن «مفاعیلن» آمده، بنابراین وزن یال مرتبط با «فعلاتن» دو برابر وزن یال مرتبط با «مفاعیلن» (در جایگاه اول) است. البته درهشت وزن عروضی انتخابی، تنوع ارکان آنقدر زیاد نیست، بنابراین بیشتر وزن‌های اولیه گراف برابر با یک هستند. نودهای انتهایی با وزن یک به خودشان بازمی‌گردند. زیرا در این مرحله ما به مقصود خود یعنی تشخیص وزن رسیده‌ایم و ادامه مسیر مطلوب ما نیست.

ما نیاز به یک نود تله یا نخودی نیز داریم. این نود برای آن است که در صورتی که بیت ورودی به هیچ یک از هشت وزن شباهت نداشت، در آن نود بمانیم. بنابراین از تمامی نودها به جز نودهای انتهایی (۲۸ تا ۳۵) به نود تله اتصال داریم و از نود تله تنها به خودش و با وزن ۱ ارتباط برقرار است. وزن یال‌های متصل به یال تله با آزمایش بر روی داده‌های موجود مقدار ۰.۲ تعیین شده است. نود تله را نود صفر می‌نامیم. وزن‌های مشخص شده بر روی یال‌ها در این مرحله را وزن اولیه نامیده و با نام $W_{i,j}(s)$ معرفی می‌کنیم. $W_{i,j}(s)$ به معنای وزن اولیه یال جهت‌دار از نود i به نود j است. توجه کنید که مجموع وزن یال‌های خروجی از یک نود باید برابر با یک باشد.

۶.۳. وزن پویای گراف

وزن پویای گراف با ورود هر نمونه (رشته صفر و یک معرف یک مصرع شعر) ساخته می‌شود. با ورود هر نمونه آن را به بخش‌های چهار نویسه‌ای تقسیم‌بندی می‌کنیم. دلیل این کار هم میانگین طول ۳.۷۵ ارکان الگوهای هشت‌گانه است. سپس شباهت هر بخش را با ارکان موجود می‌سنجیم. ما برای سنجش شباهت از الگوریتم [13] Levenshtein استفاده کردیم. البته لازم نیست شباهت هر جایگاه با تمامی ارکان مقایسه شود، تنها کافی است شباهت جایگاه اول (چهار نویسه اول نمونه ورودی) با پنج رکنی که نود یک به آنها متصل است (سطح یک گراف) مقایسه شود. همینطور برای جایگاه‌های بعدی نیز شباهت با رکن‌های سطوح معادل بررسی می‌شود. وزن‌های مشخص شده در این مرحله را وزن پویا نامیده و با نام $W_{i,j}(d)$ معرفی می‌کنیم. $W_{i,j}(d)$ به معنای وزن پویای یال جهت‌دار از نود i به نود j است. توجه کنید که وزن پویای نودهای نهایی به خودش و وزن پویای یال‌های متصل به نود تله را برابر با یک در نظر می‌گیریم.

۶.۴. ترکیب وزن‌های اولیه و پویا

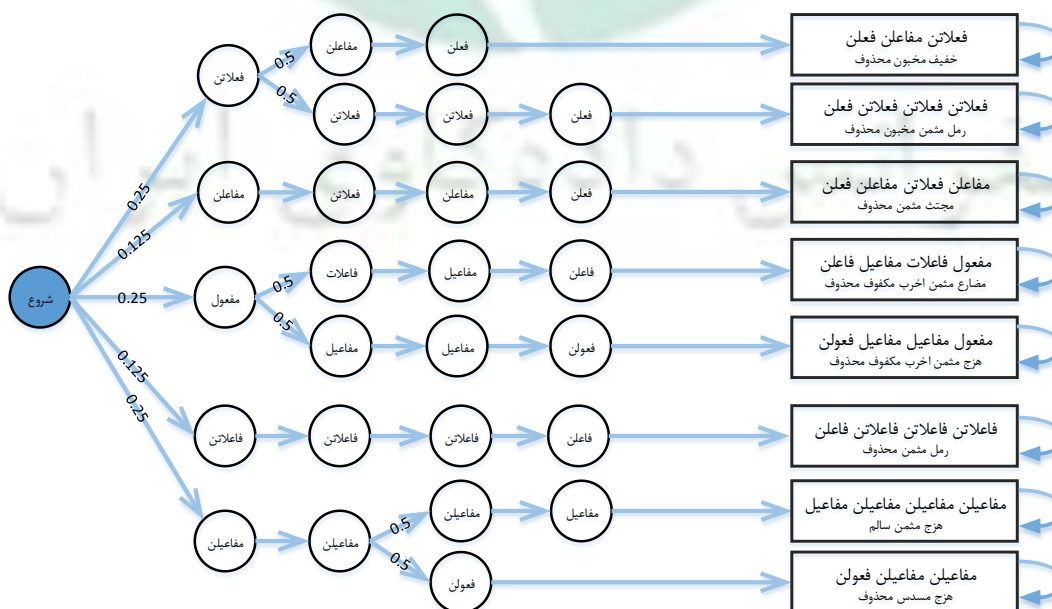
برای ترکیب وزن‌های اولیه و پویا از **Error! Reference source not found.** استفاده می‌کنیم. در این فرمول n تعداد نودها است که در گراف به دست آمده برای هشت وزن ۳۶ نود داریم (نود صفر تله است).

$$W_{i,j}(t) = \frac{w_{i,j}(s) \times w_{i,j}(d)}{\sum_{k=1}^n w_{i,k}(s) \times w_{i,k}(d)}$$

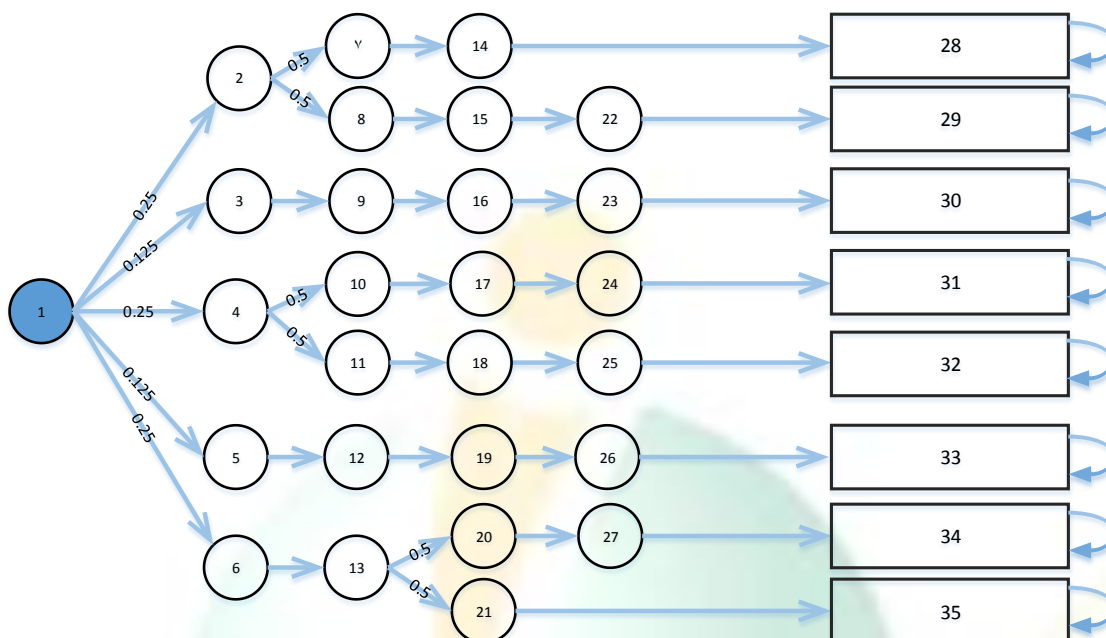
(۱) ترکیب وزن اولیه و پویا

$W_{i,j}(s)$ وزن اولیه و $W_{i,j}(d)$ وزن پویا. همچنین $W_{i,j}(t)$ وزن نهایی است. این معادله تضمین می‌کند که مجموع وزن خروجی از یک نود برابر با یک باشد. نسبت وزن پویا و اولیه نیز در این فرمول حفظ می‌شود.

شکل (۱) گراف تولید شده از ارکان موجود در هشت وزن عروضی شعر فارسی



شکل (۲) قراردادن عدد به جای ارکان در گراف تولید هشت وزن عروضی



۶.۵. بردار احتمال حالات اولیه

پس از به دست آوردن ماتریس احتمال گذار، باید بردار احتمال حالات اولیه را بسازیم. ساخت این بردار ساده است. در واقع در زمان اولیه ما به احتمال یک در نود یک هستیم. بنابراین بردار احتمال حالات اولیه ما یک بردار 36×1 تایی است که تنها عنصر یکم (عنصر صفر نیز داریم) آن یک بوده و بقیه برابر با صفر است.

۶.۶. محاسبه زنجیره مارکوف

هم اکنون تمام موارد مورد نیاز برای محاسبات زنجیره مارکوف را در اختیار داریم. با استفاده از **Error! Reference source not found.** می‌توان حالت بعدی در زنجیره مارکوف را محاسبه کرد. در این معادله Q به معنای ماتریس احتمال گذار است که در مراحل قبلی محاسبه شد و π به معنای بردار احتمال حالات است ما در مرحله قبل π_0 یا بردار احتمال حالات اولیه را به دست آوردیم. با این توصیفات به دست آوردن هر π_i ساده است.

(۲) محاسبه حالت بعدی در زنجیره مارکوف

$$\pi_i = \pi_{i-1} \times Q$$

ما ماتریس احتمال گذار را به گونه‌ای طراحی کردیم که می‌توانیم وضعیت سامانه را در حالت پایداری نیز محاسبه کنیم، زیرا در حالت پایداری تنها در نودهای انتهایی که وزن‌های ما هستند قرار داریم. اما برای سادگی محاسبات ما از همان **Error! Reference source not found.** استفاده خواهیم کرد و π_5 را محاسبه می‌کنیم.

۶.۷. کشف وزن عروضی نمونه ورودی

با به دست آوردن حالت سامانه در زمان پنجم، برداری 35 عضوی خواهیم داشت که تنها ممکن است عضوهای صفر و 28 تا 35 آن مقدار غیر صفر داشته باشند و دیگر اعضای بردار همگی صفر خواهند بود. مقدار موجود در عضو صفر نشان از احتمال عدم عضویت نمونه ورودی به الگوهای

هشتگانه دارد و مقادیر اعضای ۲۸ تا ۳۵ نیز مطابق **Error! Reference source not found.** نشان از احتمال تعلق نمونه ورودی به یکی از الگوهای وزن عروضی دارند.

کشف وزن ساده است، عضوی از بردار که بالاترین مقدار را دارد برنده است. به عنوان مثال اگر مقدار عضو سی‌ام بیشتر از بقیه اعضا باشد، وزن تشخیص داده شده «مفاعِلن فَعَلاتِن مفاعِلن فعَلن» خواهد بود. و یا اگر عضو صفر بالاترین مقدار را داشته باشد، سامانه عدم تعلق نمونه ورودی به یکی از الگوهای وزنی هشتگانه را تشخیص داده است.

۶.۸. استفاده از هر دو مصراع

واحد وزن در ادبیات فارسی مصراع است و نه بیت [1]. بنابراین ما با ورود هر بیت شعر دو فرصت برای کشف وزن داریم. در این پژوهش مراحل کشف به صورت جداگانه برای هر دو مصراع نمونه ورودی انجام شده و در نهایت بردار احتمال آنها با هم جمع شده و نتیجه مطابق بخش ۵-۷ محاسبه خواهد شد.

۷. ارزیابی زنجیره مارکوف در تشخیص وزن

پس از اجرای الگوریتم توضیح داده شده با استفاده از زنجیره مارکوف بر روی مجموعه داده موجود معیارهای سنجش دقت و بازخوانی آن مطابق **Error! Reference source not found.** به دست آمد.

جدول (۱) دقت سامانه به ازای هر وزن

وزن	Precision	Recall	F_measure	Acuracy
28	0.4912	0.9655	0.6511	0.9609
29	0.6846	0.712	0.698	0.8996
30	0.8276	0.8333	0.8304	0.9361
31	0.1897	0.0846	0.117	0.7836
32	0.1714	0.2647	0.2081	0.8214
33	0.8923	0.5979	0.716	0.94
34	0.9692	0.6632	0.7875	0.9557
35	0.4366	0.7848	0.5611	0.8735

دقت پایین زنجیره مارکوف در تشخیص برخی از اوزان قابل انکار نیست ولی در برخی دیگر دقت بسیار خوبی دارد.

۸. مقایسه نتایج

با مقایسه نتایج این پژوهش و [7] بهبود جزئی در تشخیص برخی اوزان و کاهش دقت در تشخیص برخی دیگر را شاهد خواهیم بود. مقایسه نتایج این پژوهش و پژوهش قبلی در جدول (۲) آورده شده است.

جدول (۲) مقایسه نتایج این پژوهش و پژوهش قبلی

وزن	F_measure		Acuracy	
	Markov	KNN	Markov	KNN
28	0.4538	0.6511	0.9508	0.9609

0.8996	0.897	0.698	0.5	29
0.9361	0.9038	0.8304	0.6649	30
0.7836	0.8788	0.117	0.5556	31
0.8214	0.9265	0.2081	0.4393	32
0.94	0.9167	0.716	0.5864	33
0.9557	0.9682	0.7875	0.8091	34

۹. نتیجه‌گیری و کارهای آینده

در این پژوهش ما با استفاده از پنج مرحله الگوریتم ابداعی در [7] بیت شعر ورودی را به رشته‌ای از صفر و یک تبدیل کردیم، پس از آن با استفاده از زنجیره مارکوف، وزن بیت ورودی را در یکی از اوزان هشتگانه طبقه‌بندی کردیم. مقایسه نتیجه این پژوهش و پژوهش قبلی بهبود ناچیز در دقت تشخیص در برخی اوزان را نشان می‌دهد ولی در مقابل دقت تشخیص در برخی دیگر کاهش داشته است. استفاده از هر ۳۱ وزن معروف و استفاده از کل ابیات یک غزل یا قصیده ممکن است بتواند نتایج را بهبود دهد. همچنین در این پژوهش برای ترکیب دو مصراع از جمع نتایج آنها که به صورت جداگانه به دست آمده بود استفاده کردیم، می‌توان هر دو مصراع را با هم بررسی کرد و گراف جدیدی ترسیم نمود.

۱۰. منابع

- [1] س. شمیسا، آشنایی با عروض و قافیه، تهران: نشر میترا، ۱۳۸۳
- [2] G. R. H. T. Hisar Manurung, "A Flexible Integrated Architecture For Generating Poetic Texts," The University of Edinburgh, 2000.
- [3] T. Veale, "Less Rhyme, More Reason: Knowledge-based Poetry Generation with Feeling, Insight and Wit," in Computational Creativity, 2013.
- [4] P. Gervás, "An expert system for the composition of formal Spanish poetry," Elsevier Knowledge-Baswd Systems, vol. 14, no. 3, p. 181–188, 2001.
- [5] G. P, "WASP: Evaluation of Different Strategies for the Automatic Generation of Spanish Verse," in Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, University of Birmingham, England, 2000.
- [6] G. R. H. T. Hisar Manurung, "Towards A Computational Model Of Poetry Generation," The University of Edinburgh, 2000.
- [7] م. م. مجیری و ب. مینایی بیدگلی، "تشخیص وزن عروضی اشعار فارسی: کاربرد جدیدی از متن کاوی"، کنفرانس ملی داده کاوی، تهران، ۲۰۰۸
- [8] ع. ماهیار، عروض فارسی، شیوهای نو برای آموزش عروض و قافیه، تهران: نشر قطره، ۱۳۷۴
- [9] M. M. Mojiry, "Persian Prosody," [Online]. Available: <http://www.arooz.net>. [Accessed 28 December 2013].
- [10] J. RAVIV, "Decision Making in Markov Chains Applied to the Problem of Pattern Recognition," IEEE TRANSACTIONS ON INFORMATION THEORY, vol. 3, no. 4, 1967.
- [11] M. D. B. R. K. Manuel E. Lladser, "Multiple pattern matching: A Markov chain approach," Journal of Mathematical Biology, vol. 56, no. 1-2, pp. 51-92, 2013.
- [12] "PATTERNMARKOV CHAINS: OPTIMALMARKOV CHAIN EMBEDDING THROUGH DETERMINISTIC FINITE AUTOMATA," Applied Probability Trust, vol. 45, no. 1, pp. 226-243, 2008.

[13] Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics Doklady 10, p. 707–10, 1966.



کنفرانس داده‌کاوی ایران